4D Server®

Administration für Windows und Mac OS

4D Server®
© 2005 4D SA / 4D, Inc. Alle Rechte vorbehalten

4D Server Administration
Version 2004 für Windows ® und Mac™ OS

Copyright © 1985-2005 4D SA/4D Inc.
Alle Rechte vorbehalten

Software und Handbuch unterliegen dem Copyright und dürfen weder ganz noch teilweise vervielfältigt werden. Davon ausgenommen ist die Nutzung durch den Lizenznehmer gemäß den Bestimmungen des Softwarelizenzvertrags.

4D, 4D Draw, 4D Write, 4D View, 4D Insider, 4D Server sowie 4th Dimension und das 4D Logo sind eingetragene Warenzeichen von 4D.

Windows und Microsoft sind eingetragene Warenzeichen von Microsoft Corporation.

Apple, Macintosh, Mac, ResEdit, und QuickTime sind Handelsnamen oder eingetragene Warenzeichen von Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2005, ist ein Produkt von Altura Software, Inc.

ACROBAT © Copyright 1987-2005, Secret Commercial Adobe Systems Inc. Alle Rechte vorbehalten. ACROBAT ist eingetragenes Warenzeichen von Adobe Systems Inc.

Alle anderen Produktnamen sind Handelsnamen, eingetragene Warenzeichen oder Copyrights der jeweiligen Hersteller.

WICHTIGER HINWEIS

Für die Verwendung der Software gilt der bei der Installation angezeigte Softwarelizenzvertrag. Durch Installation der Software erklären Sie sich an die Bestimmungen dieses Vertrags gebunden. Bitte lesen Sie daher den Vertragstext vollständig und genau durch, bevor Sie die Software installieren.

Inhalt

1. Einleitung 7

Überblick 9
4D Server Architektur 16

2. 4D Server in 10 Minuten 21

Die Installation prüfen 23
Eine Server Datenbank erstellen 25
Arbeitsstation an Server Datenbank anmelden 30
Die Datenbankstruktur festlegen 35
Daten mit 4D Server bearbeiten 38
Eine eigene Menüleiste hinzufügen 44
Gleichzeitig mit 4D Server arbeiten 49
4D Server ist ein Web Server 57

3. Backup 69

Backups verwalten 71
Logbuch verwalten 87
Daten wiederherstellen 96
Backup Journal 104
Backup – XML Datei 105

4. 4D Server verwalten 107

Das 4D Server Prozessfenster 109
4D Server beenden 114
Konfiguration und Cache-Speicher 116
Einstellungen zur Veröffentlichung 123
Client/Server Verbindungen verschlüsseln 125

5. 4D Client verwalten 129

Dokument mit Pfad 131
4D Client anmelden 134
Eigene Elemente übertragen (Ordner Extras) 138

6. 4D Server Menüs 139

Menü Datei/Ablage 141
Menü Bearbeiten 145
Menü Prozesse 146
Menü Daten 150
Menü Web Server 152
Menü Hilfe 153

7. 4D Programmiersprache 155

Einführung 157
4D Server und Mengen 159
Datenbankmethode On Server Startup 163
Datenbankmethode On Server Shutdown 164
Datenbankmethode On Server Open Connection 165
Datenbankmethode On Server Close Connection 171

8. Serverprozeduren 173

Serverprozeduren 175
Import mit Serverprozeduren (Beispiel) 185
Service mit Serverprozeduren (Beispiel) 188
Clients registrieren 200

1

Einleitung

4D Server ist der plattformunabhängige Daten- und Anwendungsserver für 4th Dimension im Mehrplatzbetrieb.

Dank der integrierten Architektur von 4D Server können Clients und Server eine einzige 4D Anwendung nutzen. Mit 4D Server ist es daher nicht mehr erforderlich, Client-/Server-Anwendungen getrennt voneinander zu entwickeln. 4D Server ist zudem ein Serversystem, das praktisch „wartungsfrei“ ohne Verwaltungsaufwand arbeitet: Es ist einfach zu installieren, zu verwenden und zu verwalten. Und es erlaubt die Entwicklung extrem kosteneffektiver Lösungen.

4D Server füllt die Lücke zwischen File-Sharing-gestützten Low-End-Systemen und komplexen RDBMS-Systemen auf SQL-Basis. Mit Hilfe von 4D Connectivity Plug-Ins kann jede 4D Server Anwendung problemlos mit vorhandenen Unternehmensdatenbanken genutzt werden. (Hierzu gehören beispielsweise ORACLE, Sybase oder beliebige andere ODBC-kompatible Datenquellen.) 4D Server erfüllt die Anforderungen von Arbeitsgruppen in Unternehmen beliebiger Größe und kann intuitiv und ohne besondere Vorkenntnisse verwendet werden.

Analog zu 4th Dimension Version 2004 ist auch in 4D Server ein einfach zu nutzendes Backup-System mit automatischem Logbuch integriert. Dadurch ist der sichere Betrieb gewährleistet.

Integrierte Back-end und Front-end Architektur

Mit 4D Server gibt es keinen Unterschied zwischen front-end und back-end Anwendungen. Client Software und Server Anwendung sind zwei Seiten desselben Produkts, nämlich 4th Dimension. Die 4D Server Anwendung selbst ist in zwei Teile gegliedert – 4D Server und 4D Client – diese entsprechen den Elementen der Client/Server Architektur.

4D Server, der Motor der Datenbank, liegt auf dem Server-Rechner. Er speichert und verwaltet die Datenbank auf dem Server und erlaubt Benutzern, von ihren eigenen Rechnern (Clients) aus in der Datenbank zu arbeiten.

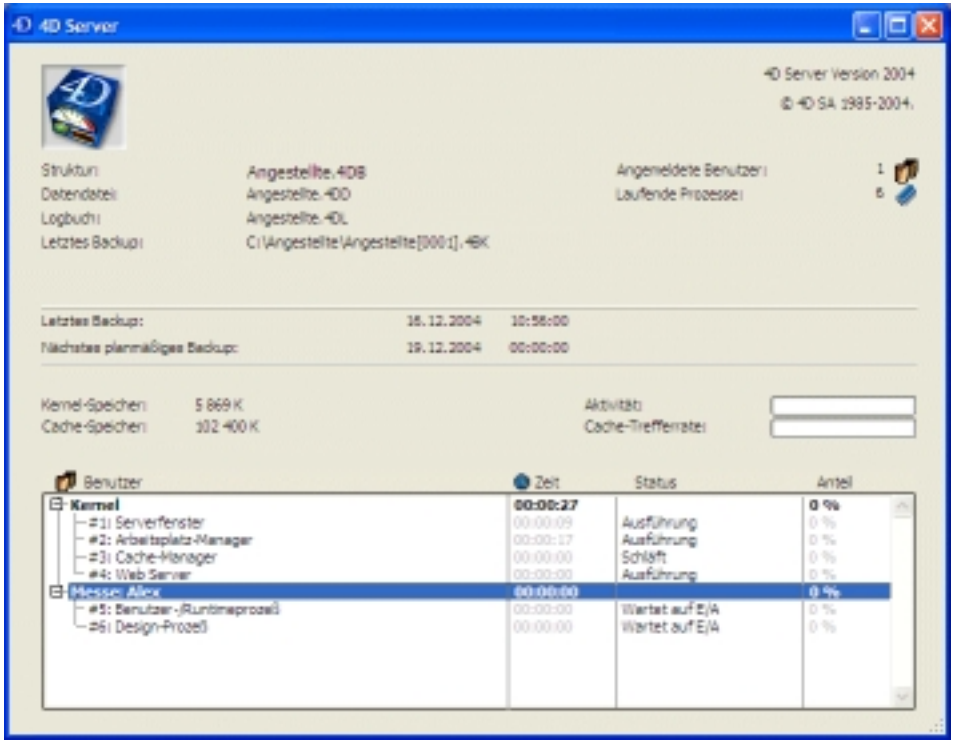
Eine Kopie von 4D Client liegt auf jedem Client-Rechner. Mit 4D Client können Benutzer auf die Datenbank auf dem Server zugreifen und verschiedene Operationen ausführen. Sie können Daten hinzufügen, Berichte erstellen und die Struktur der Datenbank verändern. Alles, was mit 4th Dimension möglich ist, kann genauso mit 4D Server und 4D Client ausgeführt werden.

Weder zusätzliche Middleware noch weitere Entwicklungsphasen sind für den Einsatz in einer Client/Server-Umgebung erforderlich. 4th Dimension, 4D Client und 4D Server nutzen dieselben Tools für die Benutzeroberfläche, dieselbe Sprache und dasselbe Datenmanagement-System.

Jede Einzelplatzanwendung, die für eine Plattform (Windows oder Macintosh) erstellt wurde, kann mühelos auf eine Client/Server-Lösung für Arbeitsgruppen skaliert werden. Genauso ist umgekehrt die Abwärtsskalierung einer mit 4D Server generierten Anwendung auf eine Einzelplatzanwendung möglich.

Wartungsfreier Daten- und Anwendungs-Server

4D Server zeichnet sich durch dasselbe hohe Maß an Benutzerfreundlichkeit wie 4th Dimension aus. 4D Server ist ein echtes Plug&Play-System.



4D Server Verwaltungsfenster

Grafisch übersichtlich zusammengefaßte Informationen: Im 4D Server Hauptbildschirm werden alle wichtigen Informationen automatisch angezeigt: der für 4D Server reservierte Speicher, der Daten-Cache, Anzahl und Name der angemeldeten Benutzer, Anzahl und Status der laufenden Prozesse, Serveraktivität, Cache-Trefferrate und Backup-Überwachung.

Automatische Konfigurierung und Skalierbarkeit: 4D Server ist für die Unterstützung neuer Clients und neuer Plug-Ins sowie für die Anbindung an andere Systeme ohne Neukonfigurierung oder Neuprogrammierung konzipiert.

Automatische und dynamische Aktualisierung sowie Versionskontrolle von Client-Workstations: Alle 4D Clients werden automatisch und dynamisch aktualisiert, wenn die Datenbank modifiziert oder ein Plug-In hinzugefügt, entfernt oder geändert wird.

Automatische asynchrone Verbindungen zu lokalen und remote Clients über das Standard Netzwerkprotokoll TCI/IP: 4D Server und 4D Client kommunizieren über das TCP/IP Protokoll transparent miteinander, unabhängig auf welcher Plattform Client und Server laufen. Das TCP/IP Protokoll ist in alle Betriebssysteme integriert, so dass keine zusätzliche Installation notwendig ist.

Automatische Verwaltung von Sitzung und Status der 4D Client- und Web-Verbindungen: Die 4D Server Datenbank-Engine generiert und verwaltet automatisch eine aktuelle Arbeitsumgebung für jede Tabelle-/Prozess-/Benutzer-Kombination. Diese sitzungsorientierte Architektur macht es möglich, dass jeder Prozess Daten für jeden einzelnen Benutzer unabhängig und simultan verwalten kann. Im kontextuellen Modus stellt der Web Server in 4D Server Web Clients mit vielen Client/Server-Funktionalitäten zur Verfügung.

Automatische Datensatzsperre: 4D Server ermöglicht das automatische Sperren und Freigeben von Datensätzen, wodurch Probleme vermieden werden, die häufig durch das Ändern gerade verwendeter Objekte entstehen. Das Sperren von Datensätzen verhindert ebenfalls Probleme, die beim Sperren von Seiten oder Dateien auftreten können.

In der Benutzeroberfläche integriertes Nachrichtensystem: Die ursprünglich desktop-orientierte 4D Server Software umfasst sämtliche Aspekte einer Bedienerführung, vereint mit modernen, integrierten Entwicklungsumgebungen. So kann 4D Server Clients über Wartungsarbeiten am Server informieren, etwa über geplante Unterbrechung von Verbindungen, Sicherungsvorgänge und vieles mehr.

Automatisierte Start- und Abmeldemethoden: 4D Server löst automatisch fünf Datenbankmethoden aus (On Server Startup, On Server Shutdown, On Server Open Connection, On Server Close Connection, On Web Connection). Die Methode On Server Startup kann automatisch alle Objekte initialisieren und laden, die möglicherweise während der Sitzung benötigt werden.

Ein unschlagbares Leistungsspektrum

4D Server bietet zusätzlich zu den Leistungsmerkmalen von 4th Dimension:

Datenmanagement im Mehrplatzbetrieb: Mehrere Benutzer können gleichzeitig in der Datenbank arbeiten. Sie können z.B. zur gleichen Zeit Datensätze hinzufügen, ändern, löschen, sortieren oder ausdrucken. 4D Server sorgt für die Datenintegrität, indem er Datensätze in Benutzung automatisch für die anderen Benutzer sperrt.

Enwicklung im Team: Entwickler können eine Datenbank einrichten bzw. modifizieren, an der andere Entwickler gerade arbeiten. 4D Server verwaltet automatisch die Daten der Arbeitsgruppen, indem er die gerade verwendeten Objekte sperrt und eine Versionskontrolle über ein Check-in/Check-out-System implementiert.

Plattformunabhängige Client /Server Architektur: 4D Server verwaltet die Datenbank für 4D Clients unter Windows und auf MacOS. Dazu gehört die simultane und plattformübergreifende Entwicklung im Team ebenso wie eine transparente Schnittstelle zu allen eingegebenen bzw. geänderten Daten auf den Arbeitsstationen, die auf verschiedenen Plattformen laufen.

4D Plug-In Architektur für Windows und MacOS: In der Windows und MacOS Version von 4D Server können Sie sowohl Plug-Ins für Windows als auch für MacOS installieren. So lassen sich plattformunabhängige 4D Plug-Ins schnell und problemlos zwischen 4D Server und 4D Client aufteilen, unabhängig von der Plattform der Arbeitsstation.

Integrierter Web Server: 4D Server und jeder 4D Client enthält eine Web Server Engine, mit der Sie 4D Datenbanken direkt im Intra- oder Internet publizieren können. Sie müssen weder ein Datenbanksystem, noch eine Web Site entwickeln, noch eine CGI Schnittstelle zwischen beiden einrichten. Ihre Datenbank ist Ihre Web Site. Sie können auch jeden 4D Client Rechner in einen Web Server umwandeln. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* im Kapitel Web Server, Überblick.

Sicherheit der Verbindung via SSL (Secured Socket Layer): Sie können 4D Server Verbindungen verschlüsseln. Die “klassische” Client/Server Architektur kann das SSL Protokoll für die Kommunikation im gesicherten Modus verwenden. Weitere Informationen dazu finden Sie im Abschnitt Client/Server Verbindungen verschlüsseln.

Trigger: Ein Trigger ist eine Methode, die einer Tabelle zugewiesen ist. Es ist demnach eine Tabelleneigenschaft. Trigger rufen Sie nicht selbst auf, sie werden automatisch von der 4D Datenbank Engine aufgerufen, immer wenn Sie Datensätze in der Tabelle bearbeiten, d.h. Datensätze laden, hinzufügen, ändern und löschen. Sie können einfache Trigger schreiben und diese dann komplexer gestalten. Trigger können “illegale” Operationen auf die Datensätze Ihrer Datenbank verhindern. Sie können mit Triggern Operationen in einer Tabelle einschränken, oder versehentliches Löschen bzw. Verfälschen von Daten ausschließen. In einem Rechnungssystem können Sie beispielsweise verhindern, dass eine Rechnung ohne Angabe des Empfängers geschrieben werden kann. Mit 4D Server werden Trigger auf dem Server-Rechner ausgeführt.

Die über Trigger festgelegten Regeln für die Datenbank gelten für alle Arbeitsstationen, egal, ob es sich um 4D Client handelt oder um eine Anwendung, die auf 4D Open basiert. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* im Kapitel Trigger.

Serverprozeduren: Sie können 4D Methoden erstellen, die in einem eigenen Prozess lokal auf dem Server-Rechner laufen. In der Client/Server Terminologie wurde dafür der Begriff "Serverprozedur" eingeführt. 4D Server dagegen bietet eine Architektur, die weit über das gängige Konzept von Serverprozeduren hinausgeht. Mit 4D Server ist eine Serverprozedur ein eigener Server Prozess, mit dem Ihr Code asynchron und unabhängig von allen anderen Prozessen auf dem Server- oder Client-Rechner läuft. In der herkömmlichen Client/Server Architektur führt eine Serverprozedur eine Anweisung aus und gibt ein Ergebnis synchron oder asynchron zurück. Mit 4D Server können Sie eine Serverprozedur starten, die während einer ganzen Client/Server Sitzung läuft und Anfragen von Clients beantwortet. Sie können auch genauso eine Serverprozedur einrichten, die nicht mit Clients interagiert, sondern über 4D Connectivity Plug-Ins oder 4D Open Daten mit einem SQL-Server oder einem anderen 4D Server abgleicht. Die Anzahl gleichzeitig ablaufender Serverprozeduren wird nur durch Hardware und Speicher begrenzt. Eine 4D Serverprozedur läuft in ihrem eigenen Prozess und kann so wie jeder andere Benutzerprozess ein individuelles Datenbankkontext haben, z.B. aktuelle Auswahlen. Die 4D Programmiersprache bietet außerdem Befehle, mit denen Client-Prozesse die Prozessvariablen jeder beliebigen Serverprozedur (einschließlich BLOB Variablen) lesen und schreiben können. Dadurch ist eine ausgeklügelte und flexible Kommunikation zwischen Clients und Serverprozeduren möglich. Schließlich können Sie mit Serverprozeduren auch neue und eigene Dienste in 4D Server hinzufügen. Weitere Informationen dazu finden Sie im Abschnitt Serverprozeduren.

Auf Client ausgeführte Serverprozeduren: Mit 4D Server können Sie von einem Server- oder Client-Rechner aus Serverprozeduren auf einem oder mehreren anderen Client-Rechnern ausführen. So können Sie die Verteilung des Arbeitsaufkommens optimieren oder Anwendungen erstellen, welche die Kommunikation zwischen den Clients ermöglicht. Weitere Informationen dazu finden Sie im Abschnitt *Serverprozeduren*.

Server Pfad: Sie können den Pfad auf eine Serverdatenbank mit einem Benutzerkennwort im Kennworteditor sichern. Der Benutzer erhält so durch einfaches Anklicken des Icons für den 4D Client Pfad Zugriff auf die Serverdatenbank. 4D Client greift über diesen Pfad automatisch auf die richtige Datenbank zu.

4D Open: Mit 4D Open, dem API (Application Programming Interface) für 4D Server können Benutzer über "fremde" Clients auf 4D Server Datenbanken zugreifen. Dies können sowohl Windows- als auch Macintosh-Anwendungen sein. 4D Open kann in 4th Dimension und 4D Client eingebunden werden. Benutzer können sich auch gleichzeitig an mehrere Server anschließen, so dass auch Systeme möglich sind, die auf mehrere Abteilungen oder auch Firmenstandorte verteilt sind. Wenn Sie 4D Open innerhalb von Triggern und Serverprozeduren einsetzen (also Methoden, die auf dem Server-Rechner ausgeführt werden), können Sie auch Systeme schreiben, bei denen sich 4D Server an andere 4D Server anbindet, um beispielsweise Daten automatisch zu reproduzieren oder verteilen. Kurzum, mit 4D Open wird 4D Server zum Client seiner eigenen Architektur.

Integriertes Backup System: 4D Server enthält ein komplettes Backup-Modul mit automatischer Restore-Funktion. Damit können Sie eine Datensicherung im laufenden Betrieb ausführen, ohne die Anwendung zu beenden. Backups lassen sich in regulären Zeitabständen und ohne Einwirken des Benutzers automatisch oder manuell starten. Sie können auch festlegen, dass die Datenbank im Störfall automatisch wiederhergestellt bzw. neu gestartet wird.

Logbuch mit Auto-Restore: In der Praxis ändert sich der Datenbestand zwischen zwei Backups. Je mehr Personen mit einer Anwendung arbeiten, desto ärgerlicher ist ein Verlust. Durch die Logbuchfunktion des Backup wird sichergestellt, dass alle Änderungen am Datenbestand (automatisch) in einem Logbuch protokolliert werden. Kommt es z.B. zu einem Festplattenausfall, können Sie mit dem letzten Backup und dem aktuellen Logbuch den Stand vor dem PlattenCrash wiederherstellen.
Wird eine Anwendung nicht ordnungsgemäß beendet, z.B. durch einen Stromausfall, sind alle Änderungen seit dem letzten Speichervorgang des Caches verloren. Stellt 4D / 4D Server beim nächsten Start fest, dass Einträge im Logbuch noch nicht im Daten-File enthalten sind, werden diese intergriert.
Noch problematischer ist die Situation, wenn der Ausfall gerade dann passiert, wenn die Daten auf Platte gesichert werden. Das Resultat ist ein inkonsistenter, bzw. beschädigter Datenbestand. Selbst dieser „worst case“ ist für das integrierte Auto-Restore kein Problem. Sollte das Daten-File defekt sein, wird das letzte Backup wiederhergestellt und dann das Logbuch eingearbeitet – auf Wunsch sogar automatisch!

Web Services anbieten und nutzen: Um eine 4D-Anwendung als SOAP-Server nutzen zu können, muss der Entwickler kaum Änderungen am bestehenen Code machen:

Mit nur einem Klick lassen sich bereits vorhandene Datenbankmethoden in publizierbare SOAP-Methoden umwandeln. Dabei erstellt 4D auf Wunsch auch die externe Dokumentation des gerade freigegebenen Dienstes in Form einer WSDL-Datei (Web Service Description Language). 4D 2004 unterstützt für Ihre eigenen Web-Services die beiden Typen RPC und DOC. Damit Sie eigene Web-Services anbieten können, muss der 4D-Web-Server aktiviert sein.

Um einen „fremden“ Web Service richtig aufrufen zu können, analysieren Sie die entsprechende WSDL-Datei. Der 4D Web Service-Assistent erledigt das für Sie und generiert automatisch eine Proxy-Methode, mit der Sie den Dienst dann nutzen können. Somit reduziert sich die Entwicklungszeit auf ein Minimum.

Update im Mehrplatz-Betrieb: Bei einem Funktions-Update in der Client/Server Installation
wird eine neue Struktur nur auf dem 4D Server eingespielt. Ist es eine neue, geänderte Strukturdatei der Anwendung, werden alle 4D Clients automatisch beim nächsten Zugriff mit den geänderten Objekten „versorgt“. Dieser automatische 4D-Update Mechanismus erlaubt Ihnen bzw. den Kunden in kürzester Zeit, eine neue Version Ihrer Anwendung einzusetzen.

In 4D 2004 ist das automatische Update der 4D Clients möglich. Diese können zentral am Server installiert werden. Technisch erzeugen Sie mit „Anwendung erstellen“ ein Client/ Server Paar Ihrer Applikation. Dieser Client wird auf den Arbeitsstationen installiert und damit gearbeitet. Spielen Sie nun auf dem Server-Rechner neue Versionen ein, z.B. basierend auf der Version 4D 2004.x, wird der bereits installierte Client automatisch aktualisiert.

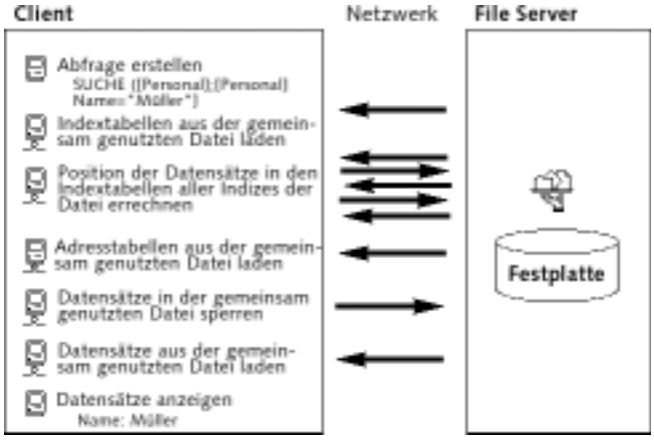
Connectivity Plug-Ins: Mit den 4D Connectivity Plug-Ins 4D ODBC Pro, 4D for OCI, 4D for MySQL 2004, 4D for PostgreSQL 2004, 4D for ADO und 4D JDBC für 4D Server können 4D Server und 4D Client direkt auf andere Systeme zugreifen, wie z.B. ORACLE, Sybase SQL Server, sowie auf jeden anderen Datenbankserver, der auf ODBC basiert. 4D Anwendungen können Daten von mehreren Servern gleichzeitig abfragen, zusammenstellen und das direkt in 4D oder über jeden beliebigen Web Browser, der auf 4D zugreifen kann.

Die 4D Server-Architektur ist leistungsstark, weil sich Arbeitsstationen und Server die Arbeit teilen.
Der Server speichert und verwaltet die Daten. Die Arbeitsstationen schicken dem Server Anfragen und erhalten von diesem die Ergebnisse.
Diese Architektur unterscheidet sich von den herkömmlichen Multiuser-Architekturen, denn sie benutzen Dateien gemeinsam (File-Sharing Prinzip)
Die 4D Server-Architektur gleicht den Client/Server-Architekturen, die mit vernetzten Stationen arbeiten. Es gibt jedoch zwei Unterschiede: Die grafische Client/Server-Architektur und die integrierte Client/Server-Architektur von 4th Dimension.
Das folgende Kapitel beschreibt die verschiedenen Möglichkeiten von 4D Server.

Arbeitsweise File Sharing (nicht bei 4D Server)

Alle Benutzer können auf dieselbe Datei zugreifen. Da aber kein Verwaltungsprogramm existiert, muss jeder Benutzer einer Arbeitsstation die Daten aufrufen, bearbeiten und dann zurückschicken.
Dieses Prinzip hat seine Grenzen. Der Datenverkehr im Netz kann eine hohe Dichte erreichen. Die Daten können nicht in einem gemeinsamen Cache-Speicher abgelegt werden. Da jede Station einen eigenen Cache-Speicher verwendet, wissen die Stationen nicht, welcher Teil des Datensatzes sich gerade bei einer Station befindet.

Folgendes Beispiel zeigt den Datenverkehr im Netz, wenn ein Benutzer in der Datenbank nach allen Personen mit dem Nachnamen Müller sucht:

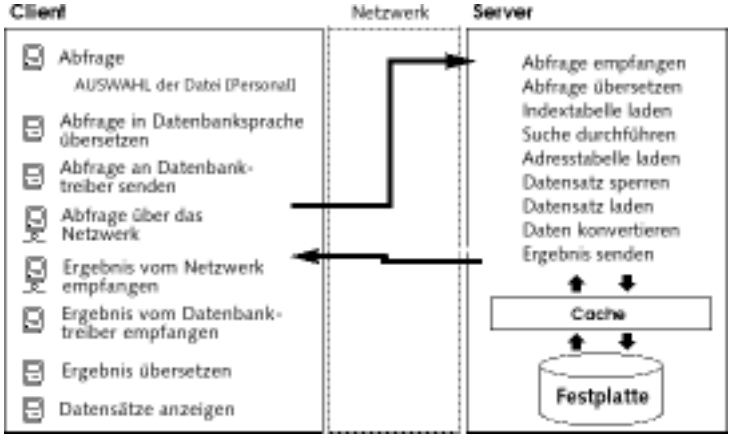


Der Server bleibt passiv, d.h., er speichert die Daten, anstatt sie aktiv zu verwalten. Er unterstützt die Benutzer nicht. Sie müssen selbst die Daten heranholen und auf der Station bearbeiten. Von daher herrscht im Netz ein ständiger Datenverkehr.

Heterogene Client/Server Architektur

Arbeitsstation und Server sind zwei vollkommen unabhängige Einheiten. Sie brauchen daher eine Kommunikationsebene, die die Daten zwischen Arbeitsstation und Server übersetzt.
Das Besondere an 4D Server ist die integrierte Client/Server-Struktur. 4D Server verwaltet natürlich auch sämtliche Daten der Datenbank grafisch und bietet alle Vorteile der Rechnerumgebung. Dazu gehören beispielsweise das einfache Installieren sowie das automatische Erkennen der verschiedenen Geräte im Netz.
4D Server und 4D Client haben die Datenstruktur von 4th Dimension. Diese sorgt für eine flexible und ökonomische Arbeitsteilung.
4D Server arbeitet sowohl mit der Daten-Engine als auch mit der Schnittstellenverwaltung von 4th Dimension.
Arbeitsstation und Server haben also auf ihrer eigenen Ebene weniger Arbeit.
Um beispielsweise nach einem Datensatz zu suchen, sendet der Client eine Anfrage an den Server. Da die Datenbank auf dem Server liegt, führt der Server den Befehl lokal auf dem Server-Rechner aus und sendet das Ergebnis an den Client-Rechner.

Folgende Abbildung zeigt den Datenverkehr im Netz, wenn ein Benutzer den Server beauftragt, nach allen Personen mit dem Nachnamen Müller zu suchen und anschließend die Datensätze anzuzeigen:



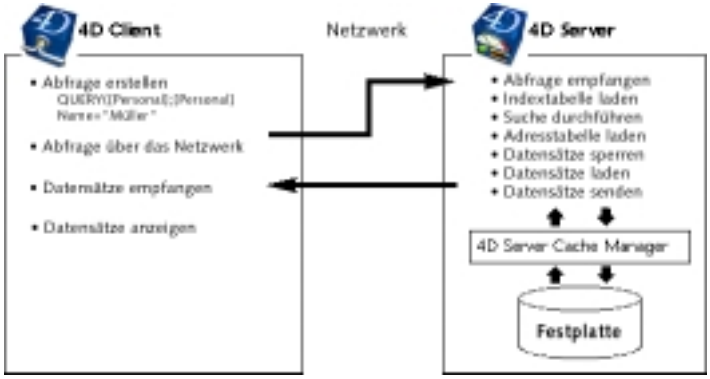
Dieses Beispiel zeigt zwei entscheidende Unterschiede zwischen File Sharing und dem Client/Server Betrieb:

• **Die Client/Server Architektur arbeitet mit Cache-Speicher**: Da nur die Engine real auf die Daten zugreift, kann der Server einen Cache-Speicher betreiben, der geänderte Datensätze speichert, bis sie auf die Festplatte geschrieben werden. Die Daten werden von einer zentralen Stelle gesendet, so dass die Arbeitsstationen immer die aktuellste Version des Datensatzes empfangen. So ist die Datenintegrität gewährleistet. Darüberhinaus werden Operationen in der Datenbank durch Verwendung des Cache-Speichers beschleunigt, da der Zugriff nicht über die Festplatte, sondern über den Cache läuft. In File Sharing Architekturen wird immer auf die Festplatte zugegriffen.

• **Datenbankoperationen im Hintergrund laufen auf dem Server ab**: Die Client/Server Architektur bietet deutlich höhere Geschwindigkeit, da Datenbankoperationen, wie Erstellen eines Index oder Sortieren von Tabellen, auf dem Server Rechner mit dessen Rechnergeschwindigkeit ausgeführt werden. In File Sharing Architekturen laufen die gleichen Operationen durch Netzwerkübertragungen und Grenzen der Arbeitsstation viel langsamer ab.

Zusammenarbeit von Arbeitsstation und Server

Bei herkömmlichen Client/Server-Architekturen besitzt der Server keine Informationen über die Umgebung der Arbeitsstation. Die Daten vom Server sind Rohdaten, die die Arbeitsstation interpretiert, z.B. mit übergebenen Arrays.
Bei 4D Server arbeiten Server und Arbeitsstation mit derselben Datenstruktur. Die Daten, die beide ans Netz senden, sind bereits strukturiert. Das heißt, Arbeitsstation und Server können sie direkt lesen.



Die Arbeitsstation ist zuständig für:
• **Anfragen**: 4D Client sendet Anfragen an den 4D Server. Sie können Anfragen über die integrierten Such- und Sortiereditoren erstellen oder über Programmierung. 4D Client bietet Editoren zum Erstellen und Ändern von Methoden und verwaltet Variablen und Arrays.
• **Antworten**: 4D Client empfängt Antworten vom 4D Server und aktualisiert die Daten auf der Arbeitsstation. Fragt der Client z.B. nach allen Datensätzen mit dem Nachnamen "Müller", erhält er diese Datensätze vom Server und zeigt sie in einem Formular an.

Der Server ist zuständig für:
• **Zugriffe**: 4D Server verwaltet über Multitasking konkurrierende Zugriffe und alle Prozesse, die von Clients erzeugt werden.
• **Struktur und Datenobjekte**: 4D Server speichert und verwaltet sämtliche Objekte der Datenbank, d.h. Daten (Datensätze, aktuelle Auswahl) bzw. Strukturobjekte (Tabellen, Datenfelder, Formulare, Methoden und Menüs).

• **Low-level Datenbankoperationen**: 4D Server führt Datenbankoperationen, wie Such- und Sortierläufe nach indizierten Datensätzen aus.

Beispiel:

Die Arbeitsstation fragt beim Server ein Formular ab. Ein umfangreiches Formular braucht eine bestimmte Zeit im Netz. Der Server schickt das Formular an die Arbeitsstation. Dort wird es ohne weitere Arbeit für den Server selbständig bearbeitet. Dadurch reduziert sich die Serveraktivität erheblich. So ist die Auslastungsrate des Servers für 10 Stationen, die gleichzeitig im Benutzer- und Änderungsmodus in 10 verschiedenen Formularen mit jeder Menge Bilder arbeiten, relativ gering.
Die gegenseitige Verständigung zwischen Server und Arbeitsstation ist besonders vorteilhaft beim Anzeigen von Listen und Verwalten der aktuellen Auswahl.
Der Server schickt mehrere Datensätze als Paket an die Arbeitsstation. Dieses enthält im Gegensatz zu einer herkömmlichen, heterogenen Client/Server-Architektur keinen Code. Die Arbeitsstation kann selbst die Daten zerlegen. Die Anzeigeschnittstelle wird automatisch verwaltet. Da die Engine der Arbeitsstation mit dem Server kommunizieren kann, überträgt die Arbeitsstation nur die tatsächlich benutzten Datenfelder.

Die Arbeitsteilung zwischen Arbeitsstation und Server ist eindeutig festgelegt. Sie erfolgt automatisch durch 4th Dimension. Sie benötigen dafür keine besonderen Kenntnisse. Sie kann vielmehr zu neuen Strategien für Ihre Anwendungen beitragen.

2

4D Server in 10 Minuten

Dieses Kapitel gibt eine kurze Einführung in folgende Bereiche:

- Eine Server Datenbank erstellen
- Clients an die erstellte Server Datenbank anmelden
- Eine Datenbankstruktur einrichten, inkl. Tabellen, Datenfeldern, Formularen und Methoden
- Einen zweiten Benutzer anmelden und gleichzeitig arbeiten
- Verbindung mit einem Web Browser

Zur Durchführung dieser Schritte benötigen Sie

- Einen Rechner, auf dem 4D Server installiert ist
- Zwei Rechner, auf denen 4D Client installiert ist
- Einen Rechner, auf dem ein Web Browser installiert ist.

(Das kann auch einer der Client-Rechner sein.)

Bevor Sie zum ersten Mal mit 4D Server und 4D Client arbeiten, sollten Sie Ihre Installation prüfen. Lesen Sie dazu die nachfolgenden Abschnitte.

Installierte Elemente

Haben Sie 4D Server und 4D Client in den Ordnern installiert, die das Programm standardmäßig vorgibt, liegen auf Ihrer Festplatte folgende Ordner und Dateien:

4D Server

Im Installationsordner Programme/4D wurde der Ordner 4D Server angelegt:

- Unter Windows enthält er das 4D Server Programm, die dazugehörigen Dateien und den Ordner 4D Extensions. Klicken Sie auf das Icon 4dserver.exe, um 4D Server zu starten.
- Auf MacOS enthält er das 4D Server Software Paket. Um 4D Server zu starten, klicken Sie einfach auf dieses Paket.

4D Client

Im Installationsordner Programme/4D wurde der Ordner 4D Client angelegt.

- Unter Windows enthält er das 4D Client Programm, die dazugehörigen Dateien und den Ordner 4D Extensions. Klicken Sie auf das Icon 4dclient.exe, um 4D Client zu starten.
- Auf MacOS enthält er das 4D Client Software Paket. Um 4D Client zu starten, klicken Sie einfach auf dieses Paket.

Wie geht es weiter?

Ihre Rechner können nur über das Netzwerk miteinander kommunizieren, wenn das Netzwerkprotokoll TCP/IP konfiguriert ist.

Sind 4D Server und 4D Client korrekt installiert, gehen Sie zum nächsten Abschnitt Eine Server Datenbank erstellen. Fehlen einige der oben aufgelisteten Ordner, schlagen Sie dazu im Installationshandbuch *4D Produkte* nach.

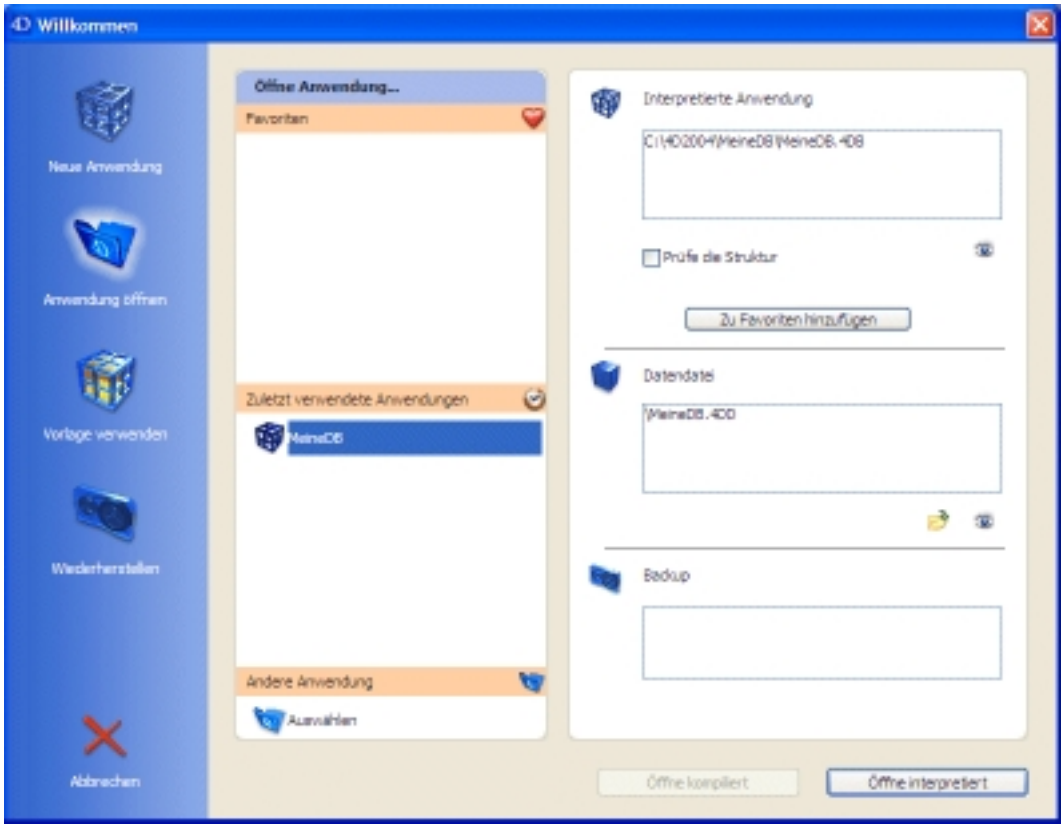
Dieser Abschnitt beschreibt, wie Sie eine Server Datenbank erstellen, auf die 4D Client bzw. ein Web Browser über das Netzwerk zugreifen können. Bevor Sie zum ersten Mal mit 4D Server und 4D Client arbeiten, sollten Sie Ihre Installation prüfen. Lesen Sie dazu im Abschnitt Die Installation prüfen nach.

Um eine 4D Server Datenbank zu öffnen, bzw. eine vorhandene zu erstellen, starten Sie 4D Server.

1. Doppelklicken Sie auf das Icon **4D Server**.

Auf dem Bildschirm erscheint der Willkommensdialog.

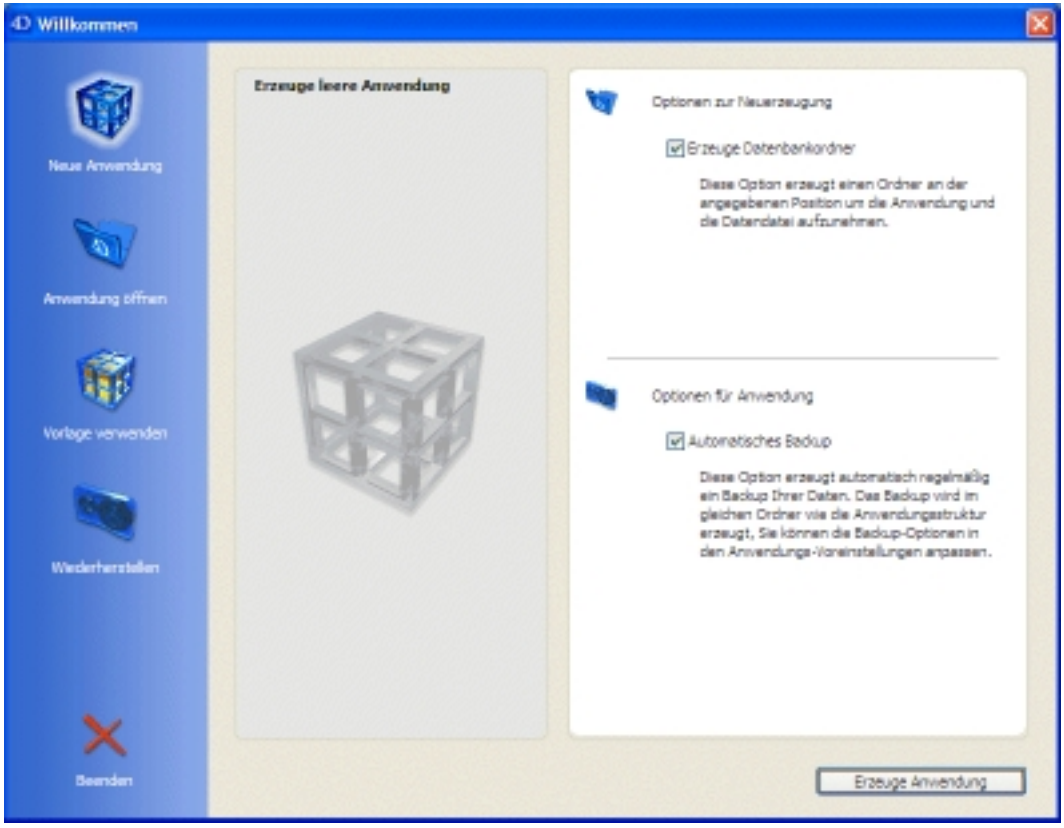
Hier können Sie eine bestehende Datenbank öffnen, eine neue einrichten oder eine archivierte Datenbank wiederherstellen.



In diesem Beispiel erstellen wir eine leere Datenbank.

2. Klicken Sie am linken Rand auf das Icon **Neue Anwendung**.

Auf dem Bildschirm erscheint folgendes Dialogfenster:


4D Willkommen
Neue Anwendung
Anwendung öffnen
Vorlage verwenden
Wiederherstellen
Beenden
Erzeuge leere Anwendung
Optionen zur Neuerzeugung
Erzeuge Datenbankordner
Diese Option erzeugt einen Ordner an der angegebenen Position um die Anwendung und die Datendatei aufzunehmen.
Optionen für Anwendung
Automatisches Backup
Diese Option erzeugt automatisch regelmäßig ein Backup Ihrer Daten. Das Backup wird im gleichen Ordner wie die Anwendungsstruktur erzeugt, Sie können die Backup-Optionen in den Anwendungs-Voreinstellungen anpassen.
Erzeuge Anwendung

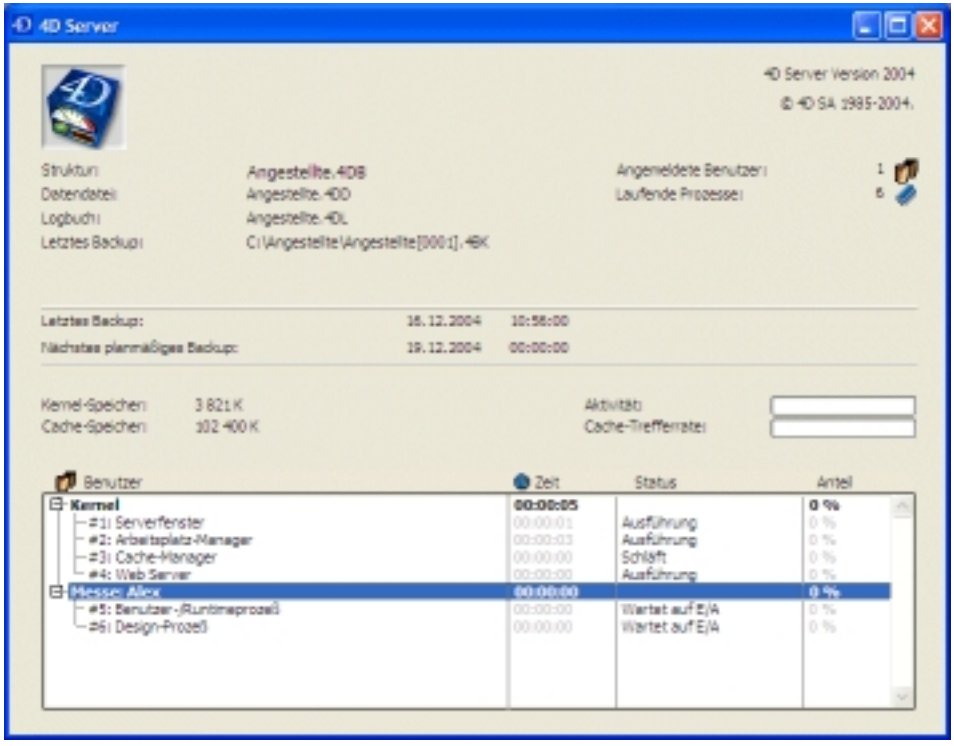
3. Klicken Sie auf die Schaltfläche **Erzeuge Anwendung**.

Auf dem Bildschirm erscheint ein Standard-Sichern Dialog, in dem Sie Name und Position der neuen Datenbank angeben.

4. Geben Sie "Angestellte" ein und klicken auf die Schaltfläche **Speichern**.

4D Server erstellt automatisch die erforderlichen Dateien und Ordner für das Operieren der Datenbank. Dann erscheint das Prozessfenster.

Es besteht aus mehreren Teilen.



Der obere Teil enthält allgemeine Angaben über den Server und die verwendeten Dateien; der mittlere Teil zeigt den Backup-Status der Datenbank und Informationen zum Speicher; der untere Teil enthält Informationen über die aktuelle Serveraktivität.

Derzeit sind keine Clients an die Datenbank angemeldet, die Anzahl der Anwender zeigt Null an. Standardmäßig werden vier Prozesse angezeigt. Diese Prozessse werden von der Datenbank-Engine (Kernel) und dem integrierten Web Server angelegt.

Der untere Bereich listet die angemeldeten Benutzer und die Prozesse auf. Sind keine Arbeitsstationen angeschlossen, wird nur die Aktivität der Server-Engine angezeigt.

Wie geht es weiter?

Die Datenbank steht nun über das Netzwerk für die Anbindung von Windows und/oder Macintosh 4D Client zur Verfügung. Sie ist jedoch noch nicht für Web Anbindungen verfügbar, da diese nicht standardmäßig zugelassen sind.

Weitere Informationen dazu finden Sie im nächsten Abschnitt Arbeitsstation an Server Datenbank anmelden. Sie lernen zuerst, wie Sie die Arbeitsstationen mit 4D Client anmelden, wie Sie die Struktur der Datenbank festlegen und einige Datensätze in der Datenbank anlegen.

Bevor Sie zum ersten Mal mit 4D Server und 4D Client arbeiten, sollten Sie Ihre Installation prüfen. Weitere Informationen dazu finden Sie im Abschnitt Die Installation prüfen.

Dieser Abschnitt beschreibt folgendes:

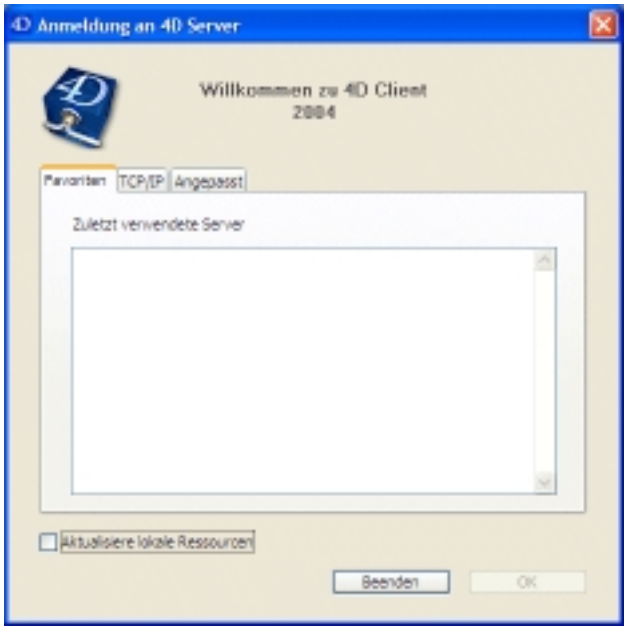
• Arbeitsstationen an die erstellte Server Datenbank anmelden.
• Eine Datenbankstruktur anlegen. Sie lernen auch, Tabellen und Datenfelder in der Datenbank zu erstellen, neue Datensätze einzugeben und bestehende Datensätze zu ändern.
• Einen zweiten Client anmelden.
• Gleichzeitig arbeiten.

Anmelden an die Datenbank

Obwohl Sie die Datenbank mit 4D Server erstellt haben (Siehe Abschnitt Eine Server Datenbank erstellen), werden alle Änderungen in der Datenbankstruktur sowie der aktuelle Datenbestand auf der Arbeitsstation angezeigt. Dieser Abschnitt beschreibt, wie Sie eine Arbeitsstation an den Server anschließen und von dort aus die Server Datenbank öffnen.

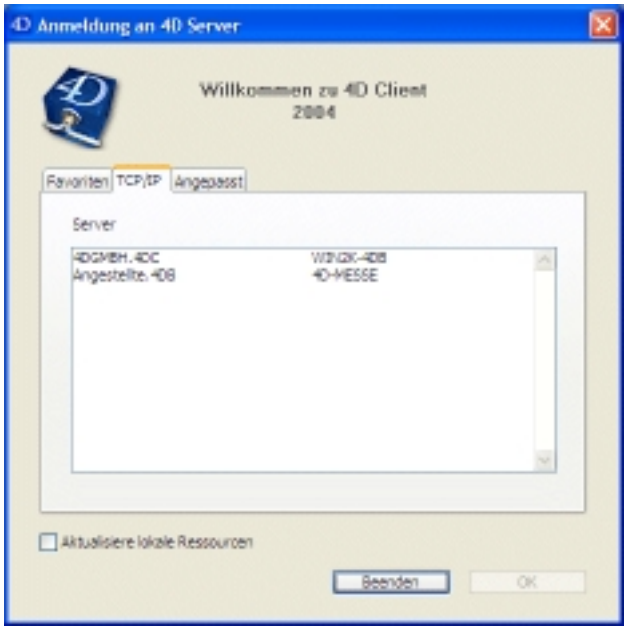
1. Doppelklicken Sie auf das Icon **4D Client**.

Auf dem Bildschirm erscheint der Anmeldedialog:



2. Klicken Sie auf die Registerkarte TCP/IP, um die Liste der über das Netzwerk laufenden Datenbanken anzuzeigen.

In der Liste erscheint die Datenbank "Angestellte":



3. Wählen Sie "Angestellte.4DB" und klicken Sie auf die Schaltfläche **OK**.

Die Datenbank wird auf der Arbeitsstation in der Designumgebung geöffnet. Sie können sofort mit der Strukturierung Ihrer Datenbank beginnen.

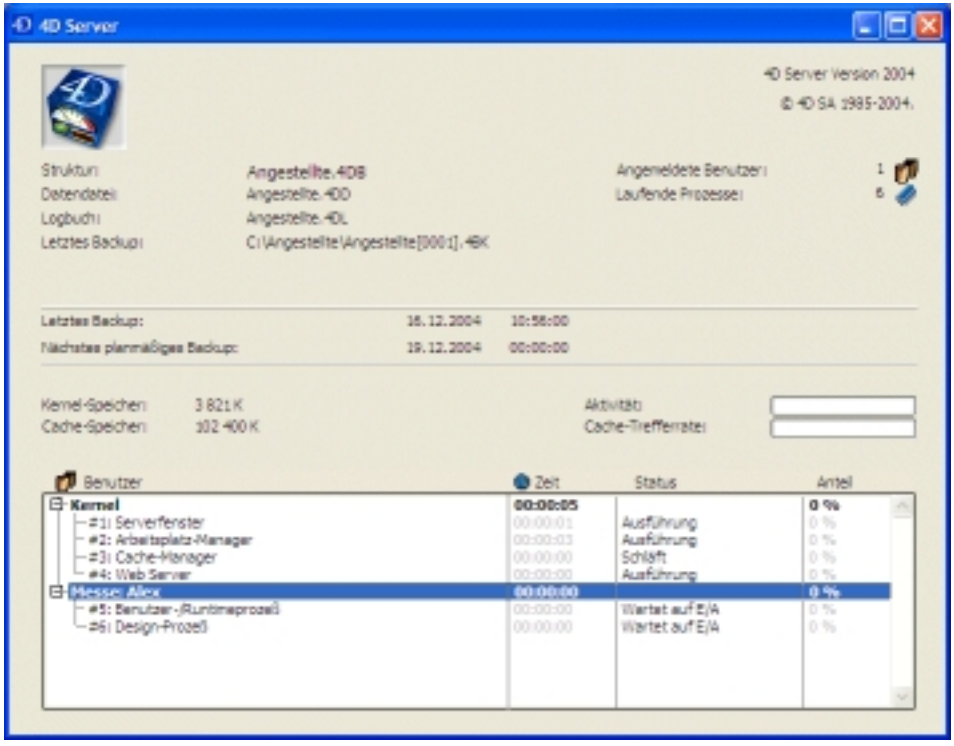
Hinweis: Erscheint nicht der Name der Datenbank, die Sie soeben mit 4D Server erstellt haben, überprüfen Sie folgende Punkte:

• Läuft 4D Server noch auf dem anderen Rechner?
• Sind Ihre Rechner an das Netzwerk angeschlossen?
• Sind die TCP/IP Einstellungen auf beiden Rechnern korrekt?
• Sind Sie noch nicht vertraut mit diesen Dialogfenstern, lesen Sie den Abschnitt Eine Server Datenbank erstellen.

Server Aktivität

Im Prozessfenster von 4D Server erscheint der Benutzername, die Anzahl der angemeldeten Anwender zeigt Eins (1) an.

Es laufen nun insgesamt sechs Prozesse:



Die ersten vier Prozesse betreffen den Server, sie wurden beim Starten von 4D Server eingerichtet. Die beiden neuen Prozesse betreffen den ersten Benutzer, der sich am Server angemeldet hat:

• Der Design-Prozess verwaltet die Designumgebung. Sobald Sie den Design-Prozess auf dem Client-Rechner über den Befehl "Struktur schließen" im Menü "Datei/Ablage" schließen, wird der Design-Prozess auf dem Server beendet.

• Der Benutzer-/Runtimeprozess verwaltet die Benutzer- und die Runtime-Umgebung. Jeder weitere Benutzer fügt in der Prozessliste mindestens zwei Prozesse hinzu. Wollen Sie die Prozesse eines Benutzers ausblenden, klicken Sie unter Windows auf das Kreuz, auf Macintosh auf den Pfeil links neben dem Benutzernamen. Wollen Sie die Prozesse eines Benutzers wieder einblenden, klicken Sie erneut unter Windows auf das Kreuz, auf Macintosh auf den Pfeil.

Wie geht es weiter?

Sie sind jetzt angemeldet und können mit der Datenbank arbeiten. Ihnen stehen alle Funktionalitäten, die 4th Dimension in der Einzelplatzumgebung hat, zur Verfügung. Als nächstes definieren Sie die Struktur der Datenbank. Gehen Sie dazu zum Abschnitt Die Datenbankstruktur festlegen.

Nachdem Sie an die Server Datenbank anmeldet sind (siehe Abschnitt Eine Arbeitsstation an die Server Datenbank anmelden), bringen Sie das Strukturfenster in den Vordergrund.

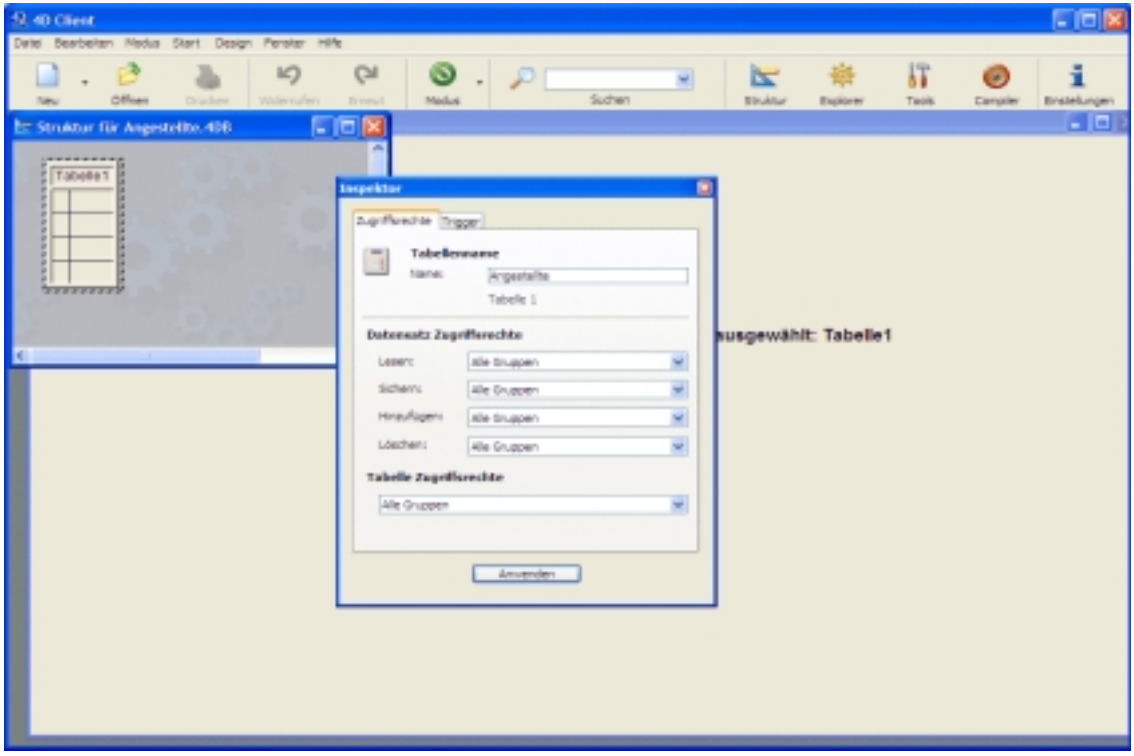
Die neue Datenbank zeigt eine Tabelle mit dem Standardnamen [Tabelle1] an.

Die Tabelle benennen

1. Doppelklicken Sie auf den Namen von [Tabelle1].
ODER
klicken Sie unter Windows mit der rechten Maustaste, auf MacOS mit gedrückter ctrl-Taste auf [Tabelle1] und wählen Sie im Kontextmenü den Befehl **Tabelleneigenschaften**.

Im Inspektor erscheint die Seite Tabelleneigenschaften.

2. Geben Sie im Bereich Tabelle den Namen "Angestellte" ein und klicken Sie auf die Schaltfläche **Anwenden**.



3. Klicken Sie im Inspektor auf die Schaltfläche **Anwenden**.

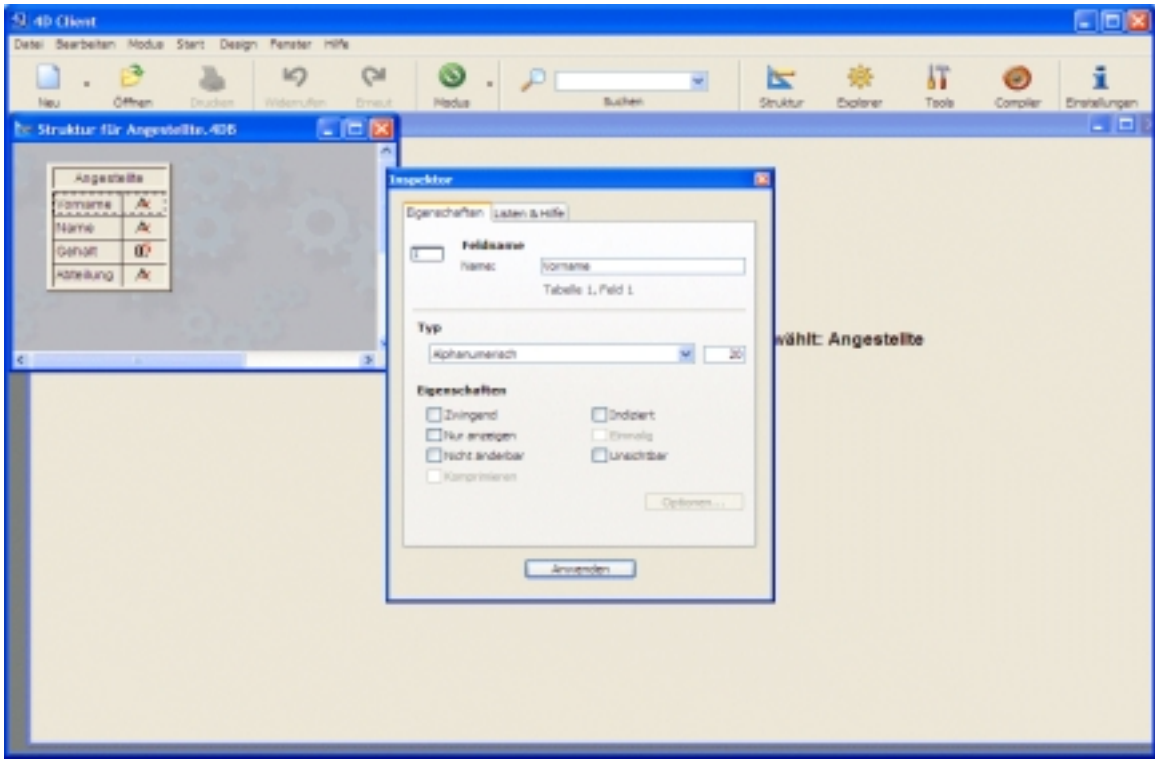
4. Klicken Sie unter Windows mit der rechten Maustaste, auf MacOS mit gedrückter ctrl-Taste auf [Tabelle1] und wählen Sie im Kontextmenü den Befehl **Neues Feld**.

Im Inspektor erscheint die Seite Feldeigenschaften.

5. Legen Sie in der Tabelle [Angestellte] folgende Datenfelder an:

Feldname	**Typ**
Vorname	Alphanumerisch (20 Zeichen)
Name	Alphanumerisch (20 Zeichen)
Gehalt	Zahl
Abteilung	Alphanumerisch (20 Zeichen)

Geben Sie für jedes Datenfeld den Namen ein, wählen Sie den Feldtyp und klicken Sie auf die Schaltfläche **Hinzufügen**.



Haben Sie alle Datenfelder festgelegt, schließen Sie das Fenster Inspektor.

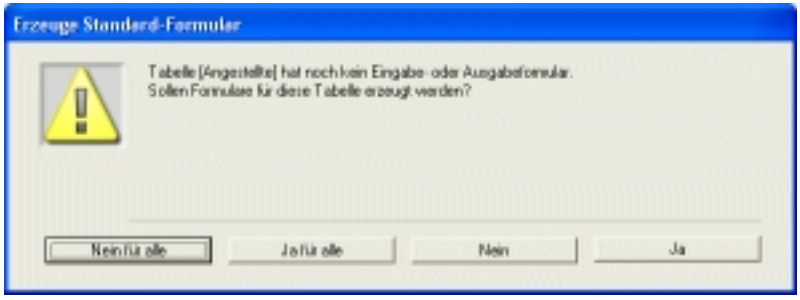
Hinweis: Arbeiten andere 4D Clients zur gleichen Zeit in der Server-Datenbank, erscheinen die Datenfelder, die Sie gerade angelegt haben, nach ein paar Sekunden auf diesen Client-Rechnern. Die Änderungen werden in Echtzeit auf den Server übertragen, erscheinen aber nicht sofort auf den anderen Bildschirmen, um zu häufiges Aktualisieren zu vermeiden.

Fomulare für die Tabelle [Angestellte] erstellen

Nachdem Sie die Tabelle [Angestellte] erstellt haben, benötigen Sie Formulare, um mit den Datensätzen arbeiten zu können. Sie können dafür den Formularassistenten verwenden und Formulare für Ihren Bedarf erstellen. 4D Server bietet jedoch praktische Tastaturkürzel für die rasche Erstellung von Ein- und Ausgabeformularen.

1. Wählen Sie im Menü **Modus** den Befehl **Benutzer**.

Sie wechseln in die Benutzerumgebung. 4D Server stellt fest, dass der Tabelle noch keine Formulare zugeordnet sind und fragt Sie, ob das Programm diese anlegen soll.



2. Klicken Sie auf die Schaltfläche **Ja**.

4th Dimension erstellt automatisch ein Eingabeformular und ein Ausgabeformular. Im Eingabeformular können Sie Datensätze einzeln hinzufügen bzw. anzeigen. Im Ausgabeformular können Sie mehrere Datensätze in einer Liste hinzufügen bzw. anzeigen.

Wie geht es weiter?

Sie können nun Daten in der Server-Datenbank eingeben und bearbeiten. Gehen Sie zum nächsten Abschnitt Daten mit 4D Server bearbeiten.

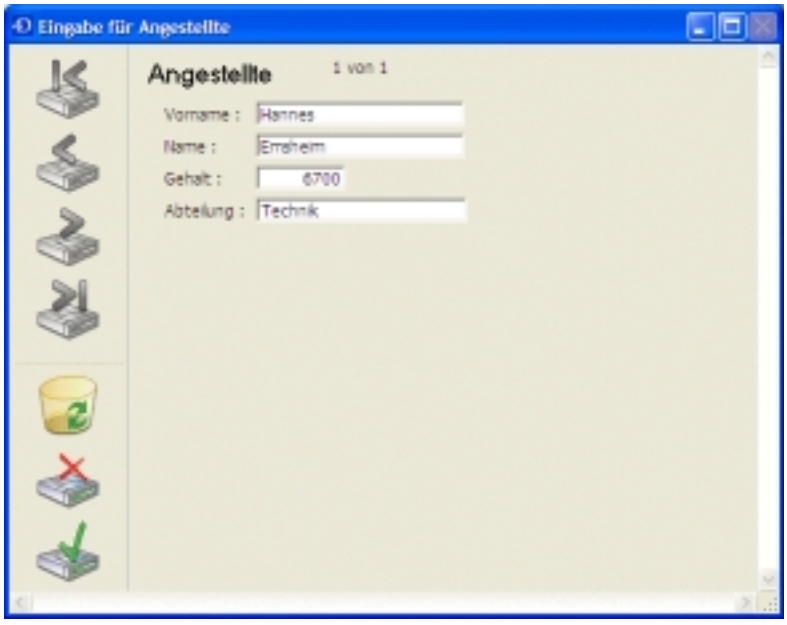
Im Abschnitt Die Datenbankstruktur festlegen haben Sie die Tabelle [Angestellte] und Standardformulare dafür erstellt. Sie können nun Datensätze eingeben.

Datensätze eingeben

In der Benutzerumgebung geben Sie Datensätze ein, suchen danach oder ändern und drucken Ihre Datensätze. Sind Sie noch nicht in der Benutzerumgebung, wählen Sie im Menü **Modus** den Befehl **Benutzer**. Auf dem Bildschirm erscheint das standardmäßige Ausgabeformular. Es wird automatisch angelegt und ist noch leer.

1. Wählen Sie im Menü **Datensätze** den Befehl **Neuer Datensatz**.
Es erscheint ein leeres Eingabeformular.

2. Geben Sie Ihren ersten Datensatz ein.
Wechseln Sie per Mausklick oder mit der Tabulatortaste von einem Feld in das nächste.



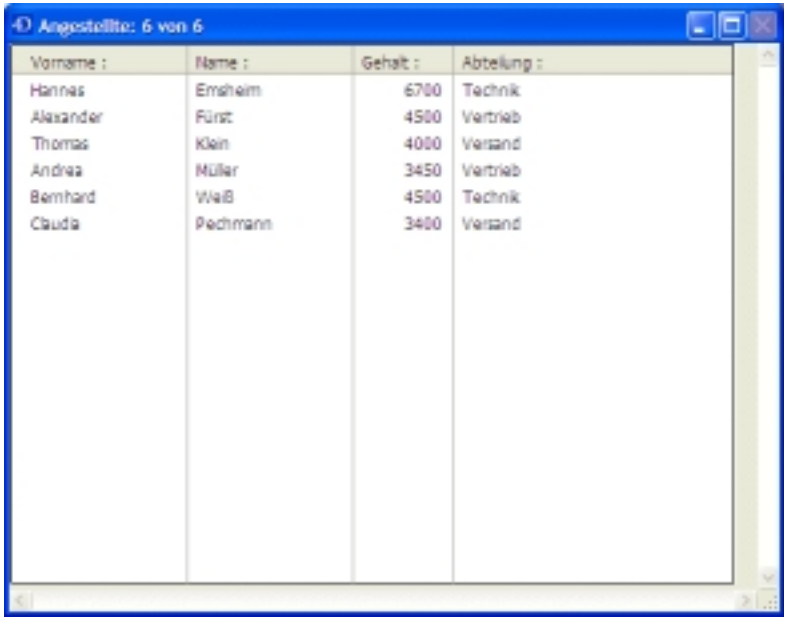
3. Klicken Sie am linken Rand auf die Schaltfläche **Bestätigen**. Der Datensatz wird gesichert.
Auf dem Bildschirm erscheint eine neues Leerformular zur weiteren Eingabe von Datensätzen.

4. Geben Sie fünf weitere Datensätze mit folgenden Werten ein:

Name	Vorname	Gehalt	Abteilung
Fürst	Alexander	4500	Vertrieb
Klein	Thomas	4000	Versand
Müller	Andrea	3450	Vertrieb
Weiß	Bernhard	4500	Technik
Pechmann	Claudia	3400	Versand

5. Haben Sie den letzten Datensatz eingegeben, klicken Sie auf **Abbrechen** (Icon mit Kreuz am linken Rand), um das neue Leerformular zu annullieren. 4D Server kehrt zum Ausgabeformular zurück.

6. Werden nicht alle sechs Datensätze angezeigt, wählen Sie im Menü **Datensätze** den Befehl **Alle anzeigen**. Ihr Bildschirm sollte folgendermaßen aussehen:



Die Datensätze sind nun in der Datenbank auf dem Server-Rechner gespeichert. Meldet sich eine zweite Arbeitsstation auf dem Server-Rechner an, kann sie die soeben eingegebenen Datensätze anzeigen. Ebenso können Sie von Ihrer Arbeitsstation aus alle Datensätze anzeigen, die in der Datenbank gespeichert sind, inkl. der Datensätze, die von anderen Arbeitsstationen eingegeben wurden. Jeder Benutzer hat Zugang zu den Datensätzen, die auf dem Server gespeichert sind.

Datensätze suchen

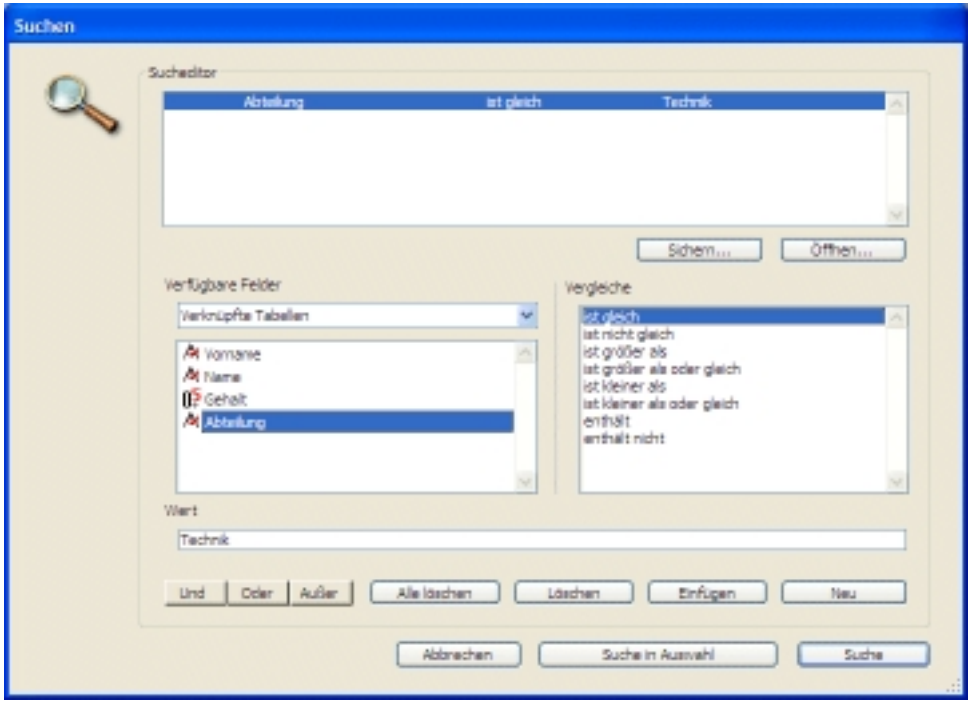
Sobald Sie Datensätze in der Tabelle [Angestellte] eingegeben haben, können Sie mit den Datensätzen arbeiten, z.B. Such- oder Sortierläufe durchführen oder drucken.
Sie können beispielsweise nach den Angestellten einer Abteilung suchen.

1. Klicken Sie in der Menüleiste auf das Icon für **Suchen**.

Auf dem Bildschirm erscheint der Sucheditor.

2. Klicken Sie im Bereich Verfügbare Felder auf "Abteilung", in der Liste Vergleiche auf “ist gleich”und geben Sie im Bereich Wert "Technik" ein.

Der Sucheditor sieht jetzt folgendermaßen aus:



3. Klicken Sie am unteren Rand auf die Schaltfläche **Suche**.

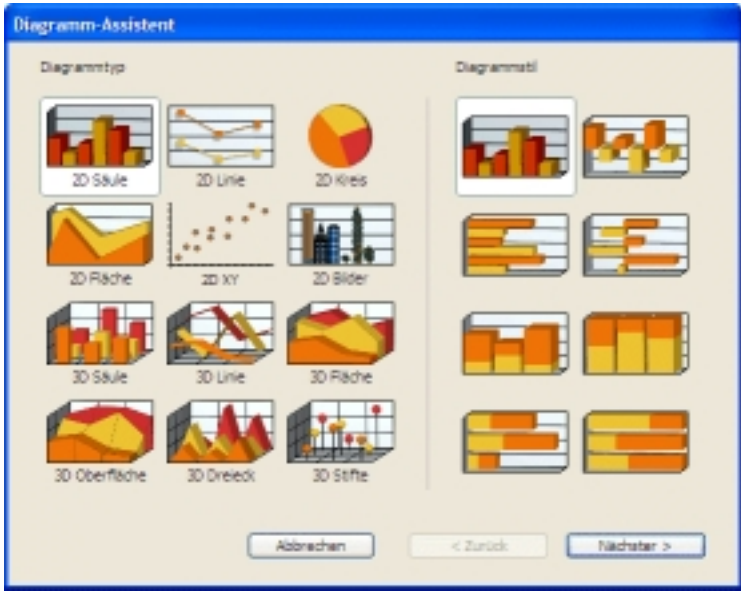
Die Anfrage wird an den 4D Server gesendet, 4D Server sendet das Ergebnis an den Client zurück. Im Ausgabeformular erscheinen nur die Angestellten der Abteilung Technik.

4. Wollen Sie wieder alle Datensätze anzeigen, wählen Sie im Menü **Datensätze** den Befehl **Alle anzeigen**.

Ein Diagramm erstellen

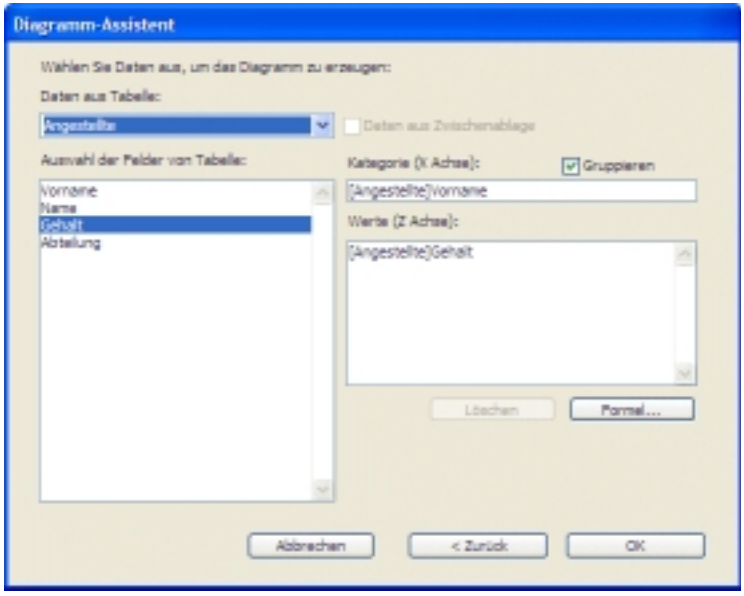
1. Wählen Sie im Menü **Bericht** den Befehl **Diagramme**.

Auf dem Bildschirm erscheint der Diagramm-Assistent:

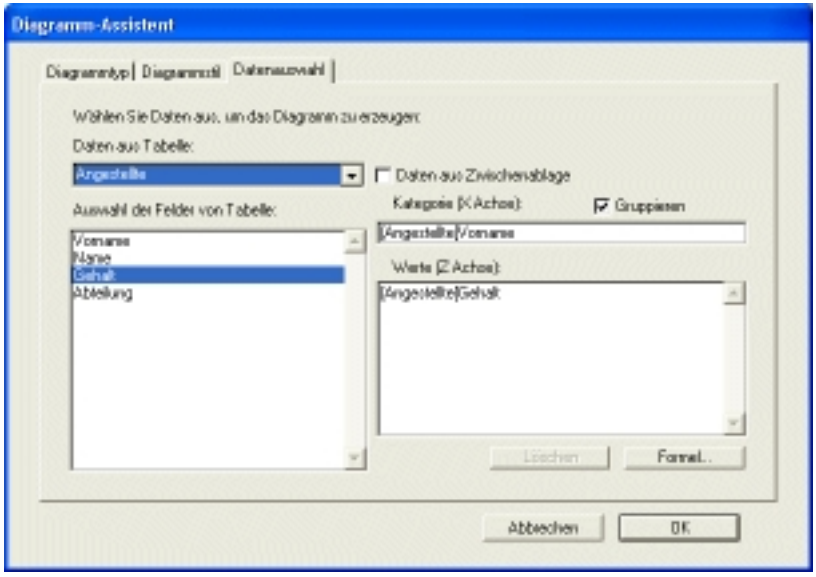


Sie erstellen ein 2D Säulendiagramm. Diese Art ist standardmäßig ausgewählt.

2. Klicken Sie am unteren Rand auf die Schaltfläche **Nächste**.



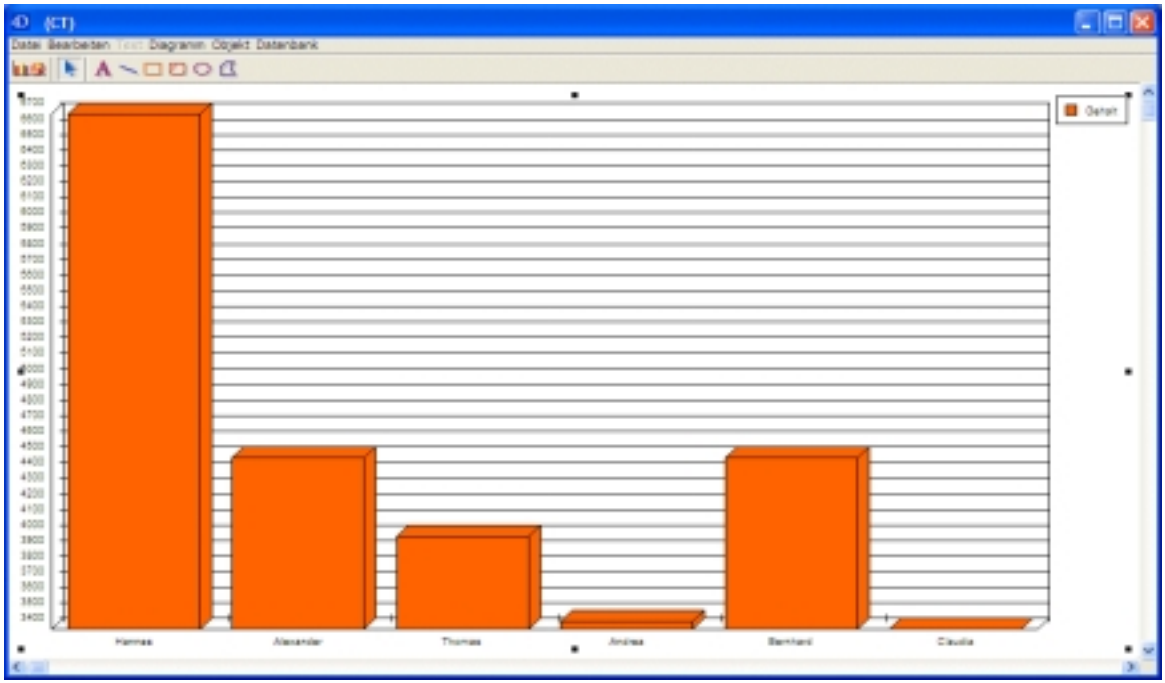
Auf dem Bildschirm erscheint das Fenster Diagramm-Assistent. Hier müssen Sie die Daten für Ihr Diagramm auswählen.



3. Ziehen Sie das Datenfeld Name per Drag & Drop in den Bereich Kategorie (X Achse).

4. Ziehen Sie das Datenfeld Gehalt per Drag & Drop in den Bereich Werte (Z Achse).

5. Klicken Sie auf die Schaltfläche **OK**.
Ihr Diagramm sieht folgendermaßen aus.



Wie geht es weiter?

Sie haben in wenigen Minuten eine Server Datenbank erstellt, eine Tabelle angelegt, Datensätze eingegeben und dann mit diesen Daten eine Suche ausgeführt und ein Diagramm erzeugt.

Sie lernen nun, eine eigene Menüleiste hinzuzufügen. Gehen Sie zum nächsten Abschnitt Eine eigene Menüleiste hinzufügen.

In diesem Abschnitt erstellen Sie zwei Methoden und eine eigene Menüleiste, d.h. Sie erstellen Ihre eigene 4D Anwendung.

Zwei Methoden hinzufügen

1. Kehren Sie in die Designumgebung zurück. Wählen Sie dazu im Menü **Modus** den Befehl **Design**.

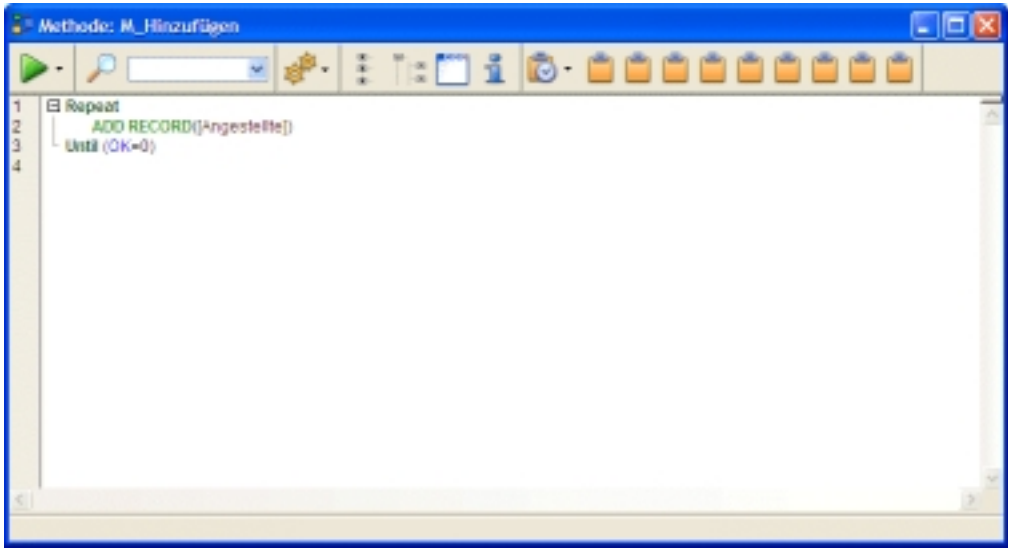
2. Wählen Sie im Menü **Datei/Ablage** den Befehl **Neu>Methode**.

Es erscheint das Dialogfenster Neue Methode.

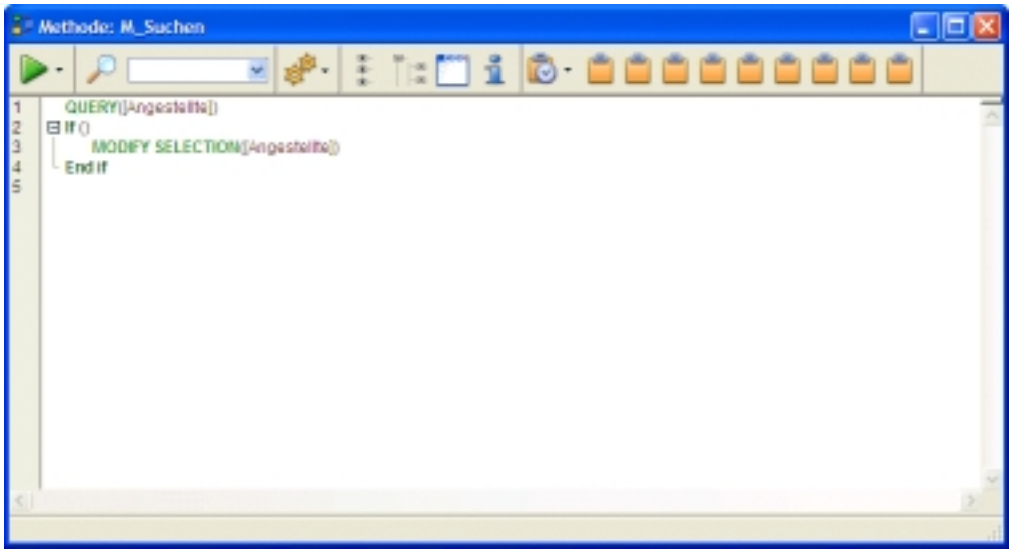
3. Geben Sie im Bereich Methodenname “M_Hinzufügen” ein und klicken Sie auf die Schaltfläche **OK**.

Auf dem Bildschirm erscheint das Methodenfenster mit dem Titel "Hinzufügen"

4. Geben Sie folgenden Code ein:



5. Erstellen Sie eine 2. Methode mit Namen “M_Suchen” und geben Sie folgenden Code ein:

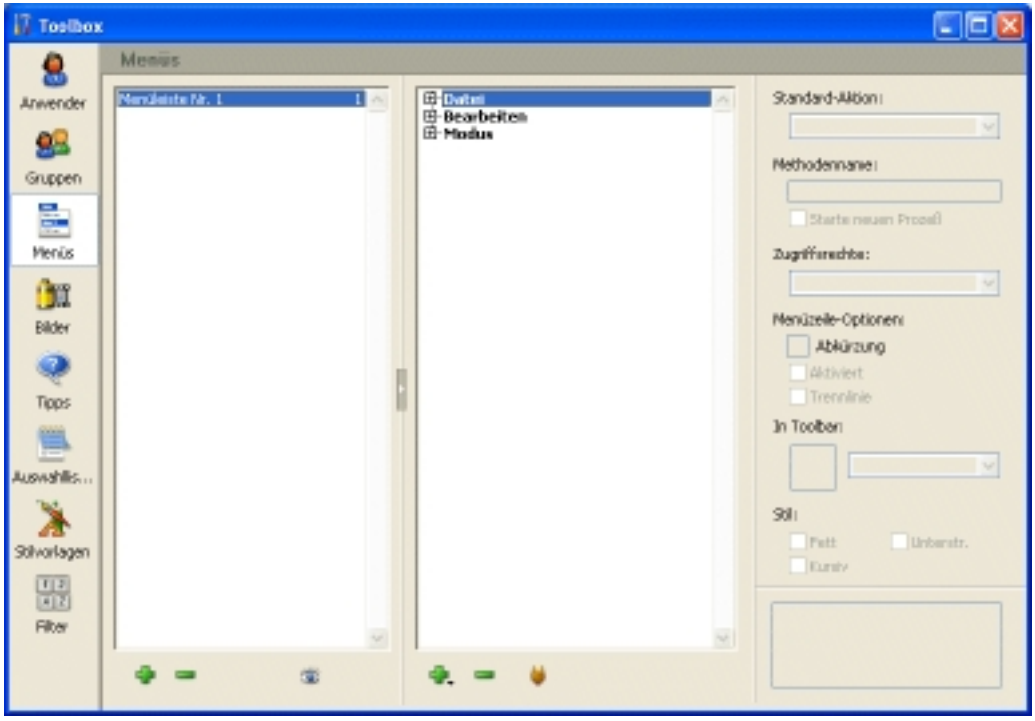


Haben Sie beide Methoden erstellt, können Sie eine eigene Menüleiste einrichten und diese Methoden Ihren eigenen Menübefehlen hinzufügen.

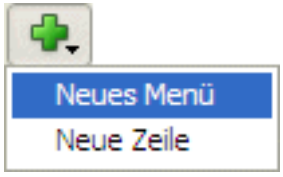
Eine eigene Menüleiste einrichten

1. Wählen Sie im Menü **Design** den Befehl **Toolbox>Menüs**.

Auf dem Bildschirm erscheint der Menüeditor mit der Standard Menüleiste.



2. Klicken Sie unter der Liste der Menüleisten auf das Pluszeichen und wählen den Eintrag **Neues Menü**.



3. Geben Sie den Menünamen “Angestellte” ein.

4. Markieren Sie den Eintrag **Neue Zeile**, um diesem Menü einen Befehl hinzuzufügen.

5. Geben Sie für den ersten Befehl im Textbereich “Hinzufügen” ein und klicken Sie auf die **Eingabetaste**.

6. Markieren Sie erneut den Eintrag **Neue Zeile**, um diesem Menü einen zweiten Befehl hinzuzufügen.

7. Geben Sie “Suchen” ein.

Ihre Menüleiste sieht folgendermaßen aus:

8. Klicken Sie auf den Menübefehl “Hinzufügen” und geben Sie im Bereich Methodenname “M_Hinzufügen” ein.

9. Klicken Sie auf den Menübefehl “Suchen”und geben Sie im Bereich Methodenname “M_Suchen” ein.

Ihre Menüleiste sieht nun folgendermaßen aus:

10. Schließen Sie das Fenster Toolbox.

Ihr eigenes Menü ist fertig!

11. Wählen Sie im Menü **Modus** den Befehl **Runtime**.

Sie verwenden jetzt Ihre eigene Anwendung mit den selbsterstellten Menüs.

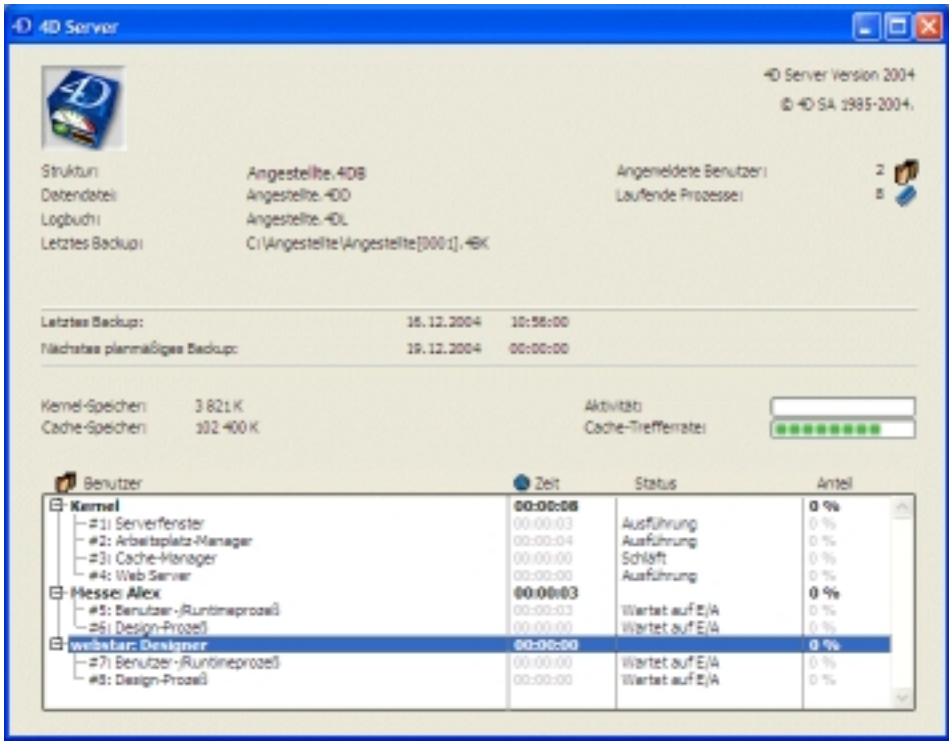
Wählen Sie beispielsweise im Menü **Angestellte** den Befehl **Suchen**, erscheint der Sucheditor (der integrierte Sucheditor aus der Benutzerumgebung). Sie können Ihre Suche definieren, die daraus resultierenden Datensätze anzeigen und ändern.

Sie haben, ohne zu wissen, soeben eine Mehrplatz-Anwendung entwickelt!
Die Erklärung dafür finden Sie im nächsten Abschnitt Gleichzeitig mit 4D Server arbeiten.

Arbeiten Sie das Lernprogramm unter Windows durch, können Sie diese Server-Datenbank ohne Änderungen auf Macintosh verwenden. Arbeiten Sie das Lernprogramm auf Macintosh durch, können Sie diese Server-Datenbank ebenfalls ohne Änderungen unter Windows verwenden.

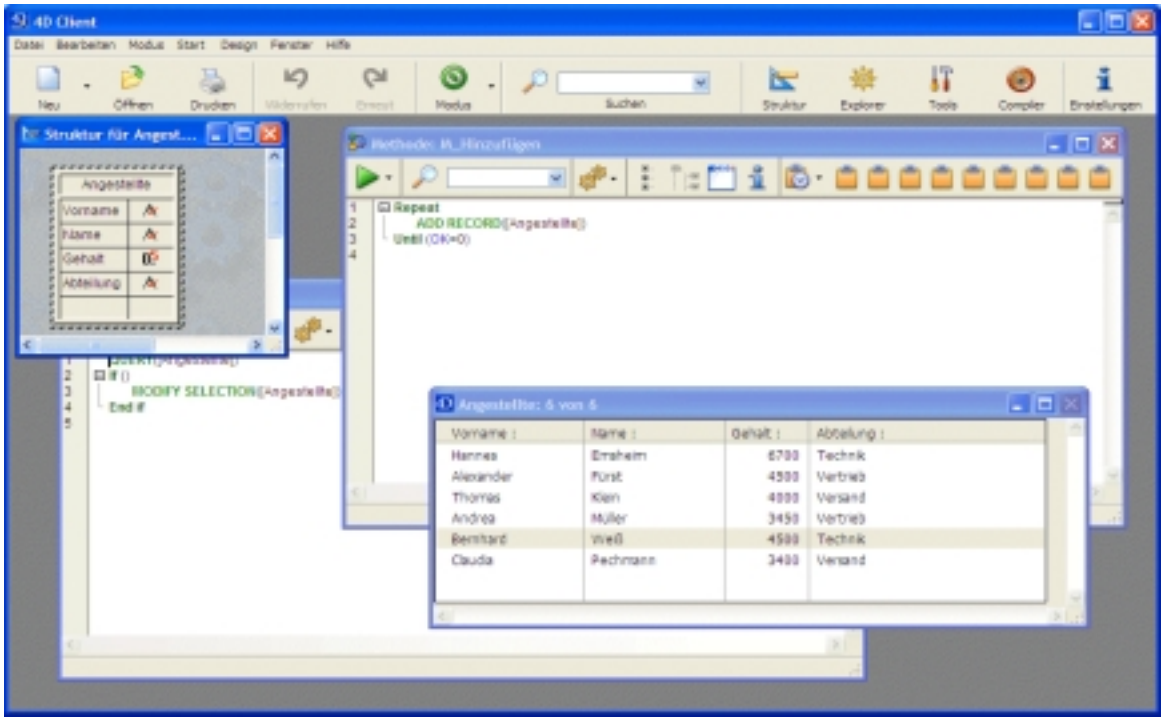
Eine zweite Arbeitsstation anmelden

Melden Sie für dieses Lernprogramm je einen 4D Client für Windows und für MacOS an. Sobald dieser angemeldet ist, wird im 4D Server Prozessfenster ein zweiter Benutzer eingetragen:



Auf jeder Arbeitsstation können Sie alles, was Sie auf der anderen Plattform eingerichtet haben, sofort verwenden.

Die Designumgebung des Windows-Client sieht folgendermaßen aus:



Alle sechs Datensätze und die beiden Methoden stehen zur Verfügung.

Gleichzeitig mit Datensätzen arbeiten

1. Wechseln Sie auf der ersten Arbeitsstation in den Runtime-Modus und wählen im Menü **Angestellte** den Befehl **Suchen**. Suchen Sie nach den Datensätzen mit dem Kriterium "Abteilung ist gleich Technik".

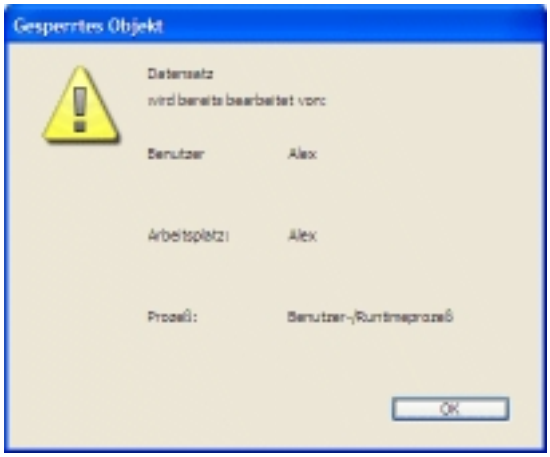
2. Wiederholen Sie denselben Vorgang für die zweite Arbeitsstation.
Sie erhalten auf beiden Rechnern dieselbe Liste.

3. Doppelklicken Sie auf der ersten Arbeitsstation auf den Datensatz “Hannes Emsheim”.

4. Wiederholen Sie denselben Vorgang für die zweite Arbeitsstation.

4D Server aktiviert den eingebauten Mechanismus zum Sperren von Objekten.

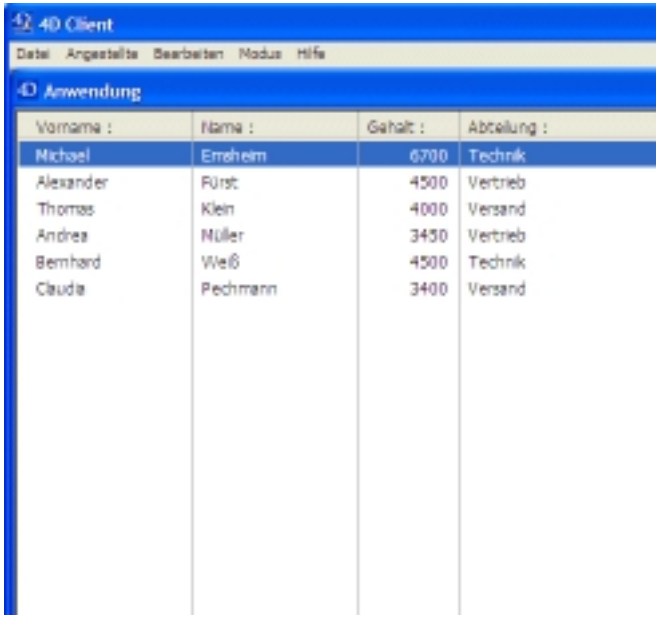
Sie erhalten die Meldung, dass der Datensatz bereits benutzt wird:



Sie können diesen Datensatz aber im Lesemodus öffnen, d.h. Sie können ihn anzeigen, jedoch nicht ändern.

5. Ändern Sie auf der ersten Arbeitsstation den Vornamen um in “Michael” und bestätigen Sie Ihre Änderungen.

Auf dem Bildschirm erscheint die aktualisierte Liste:


4D Client
Datei Angestellte Bearbeiten Modus Hilfe
4D Anwendung
Vorname : Name : Gehalt : Abteilung :
Michael Emsheim 6700 Technik
Alexander Fürst 4500 Vertrieb
Thomas Klein 4000 Versand
Andrea Müller 3450 Vertrieb
Bernhard Weiß 4500 Technik
Claudia Pechmann 3400 Versand

6. Annullieren Sie auf der zweiten Arbeitsstation die Anzeige des Datensatzes im Eingabeformular.

Auf dem Bildschirm erscheint ebenfalls die aktualisierte Liste:

Gleichzeitig mit Design Objekten arbeiten

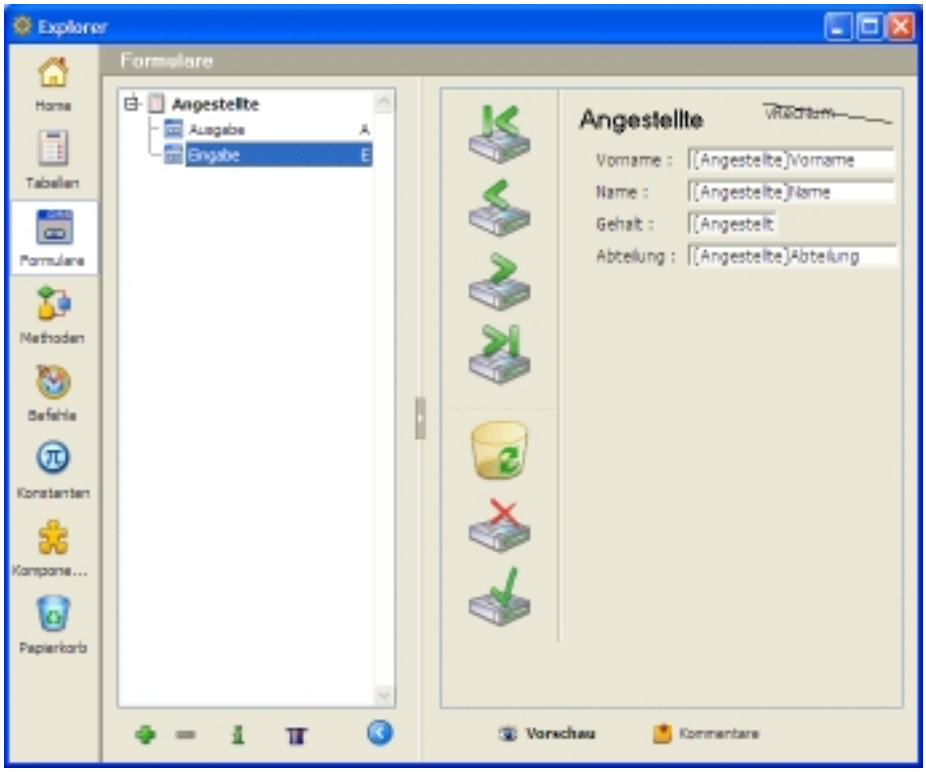
4D Server ist Daten- und Anwendungsserver zugleich. Das bedeutet folgendes:

1. Drücken Sie auf der zweiten Arbeitsstation die **Escape-Taste** und wählen Sie im Menü **Modus** den Befehl **Design**.

2. Wiederholen Sie diesen Vorgang auf dem ersten Rechner.

3. Wählen Sie auf der ersten Arbeitsstation im Menü **Design** den Befehl **Explorer>Formular**.

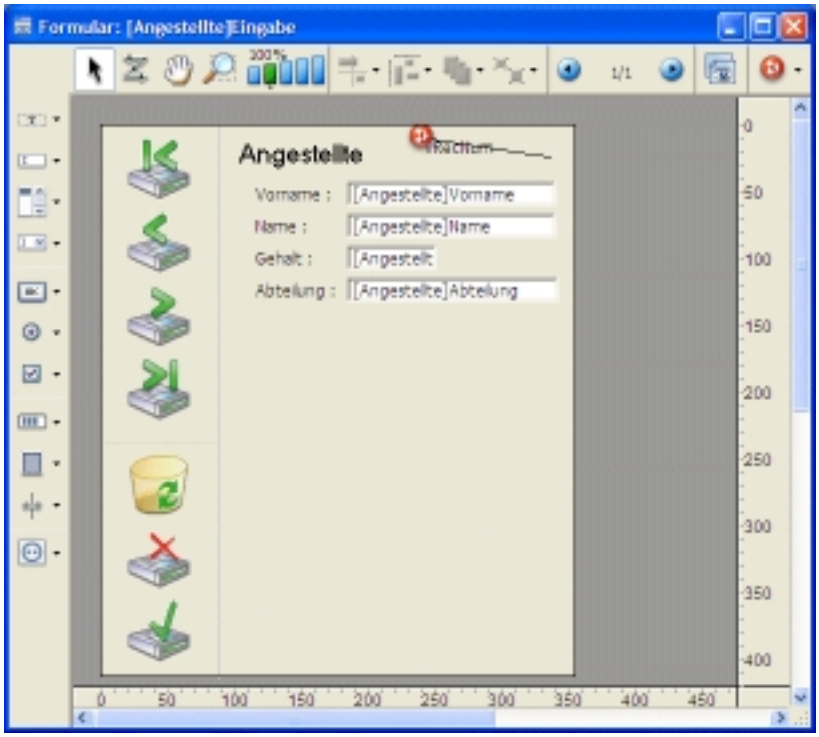
Auf dem Bildschirm erscheint das Fenster Explorer.



4. Erweitern Sie die Tabelle [Angestellte].

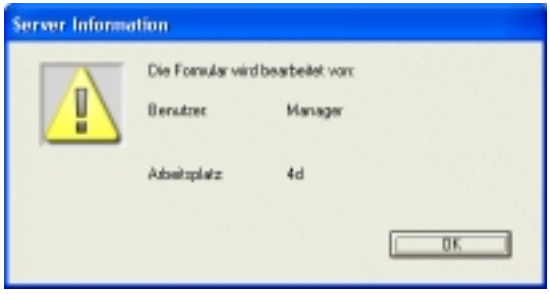
5. Doppelklicken Sie auf das Eingabeformular und klicken auf die Schaltfläche **Bearbeiten**.

Auf dem Bildschirm erscheint folgendes Fenster:



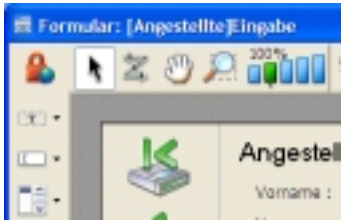
6. Wiederholen Sie diesen Vorgang auf dem zweiten Rechner.

Da dieses Formular bereits auf der ersten Arbeitsstation im Bearbeitungsmodus geöffnet ist, sperrt 4D Server diesen Vorgang. Sie erhalten folgende Meldung:



Sie können dieses Formular auf der zweiten Arbeitsstation zwar nicht bearbeiten, jedoch Objekte auswählen und in anderen Formulare einsetzen.

Das Schloss in der Ecke oben links weist darauf hin, dass Sie keine Änderungen vornehmen können.



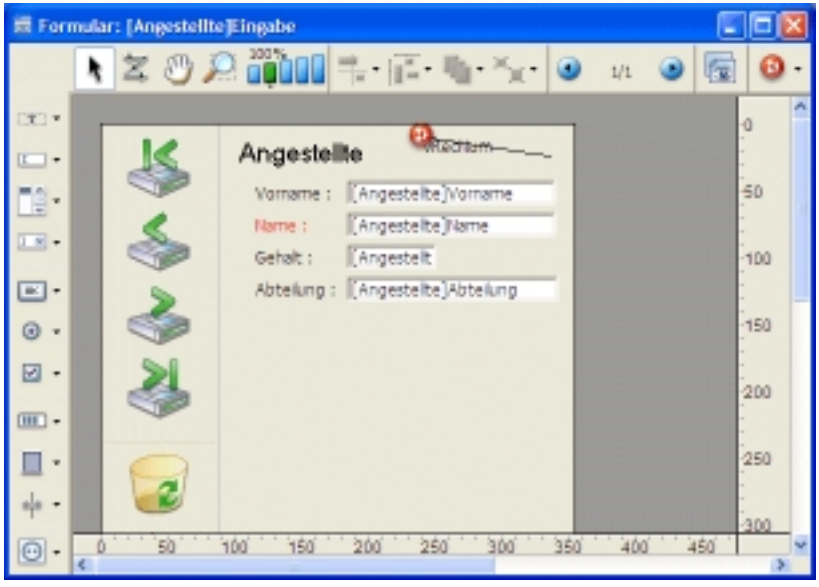
7. Wählen Sie auf der ersten Arbeitsstation die Bezeichnung "Name" links neben der Tabelle [Angestellte].

8. Wählen Sie im Menü **Objekt**>**Farbe** in der Farbpalette einen roten Farbton, um dieses Objekt rot einzufärben.

9. Wählen Sie im Menü **Datei/Ablage** den Befehl **Speichern Formular [Angestellte]Angabe**.

10. Schließen Sie das Formular auf der zweiten Arbeitsstation und öffnen es wieder, um es neu zu laden.

Auf dem Bildschirm erscheint nun das Formular mit den Änderungen, die Sie auf der ersten Arbeitsstation gemacht haben.



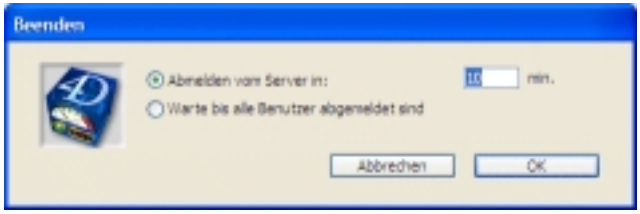
Wie Sie sehen, können mit 4D Server mehrere Benutzer gleichzeitig an derselben Datenbank arbeiten!

Den Server abschalten

4D Server informiert nicht nur die Arbeitsstationen, wenn mehrere Benutzer gleichzeitig auf denselben Datensatz oder dasselbe Objekt zugreifen wollen, sondern auch wenn der Server abgeschaltet wird.

1. Beide Arbeitsstationen bleiben an die Server-Datenbank angeschlossen. Wählen Sie auf dem Server-Rechner unter Windows im Menü **Datei**, auf MacOS im Menü **4D Server** den Befehl **Beenden**.

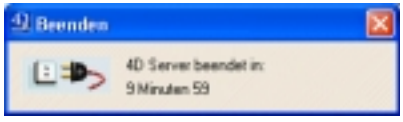
Auf dem Bildschirm erscheint folgender Dialog:



2. Klicken Sie auf die Schaltfläche **OK**.

Die beiden Arbeitsstationen erhalten unmittelbar darauf die Meldung, dass der Server abgeschaltet wird. Legt ein Benutzer gerade einen neuen Datensatz an, bleibt ihm genügend Zeit, die Eingabe zu beenden und zu sichern.

Sie können beim Abschalten des Servers auch die Option “Warte bis alle Benutzer abgemeldet sind" verwenden.



3. Während der Server abschaltet, beenden Sie 4D Client auf den beiden Arbeitsstationen.

Wie geht es weiter?

Nach diesen neun intensiven Minuten und bis der Server abgeschaltet hat, gönnen Sie sich eine Kaffeepause.

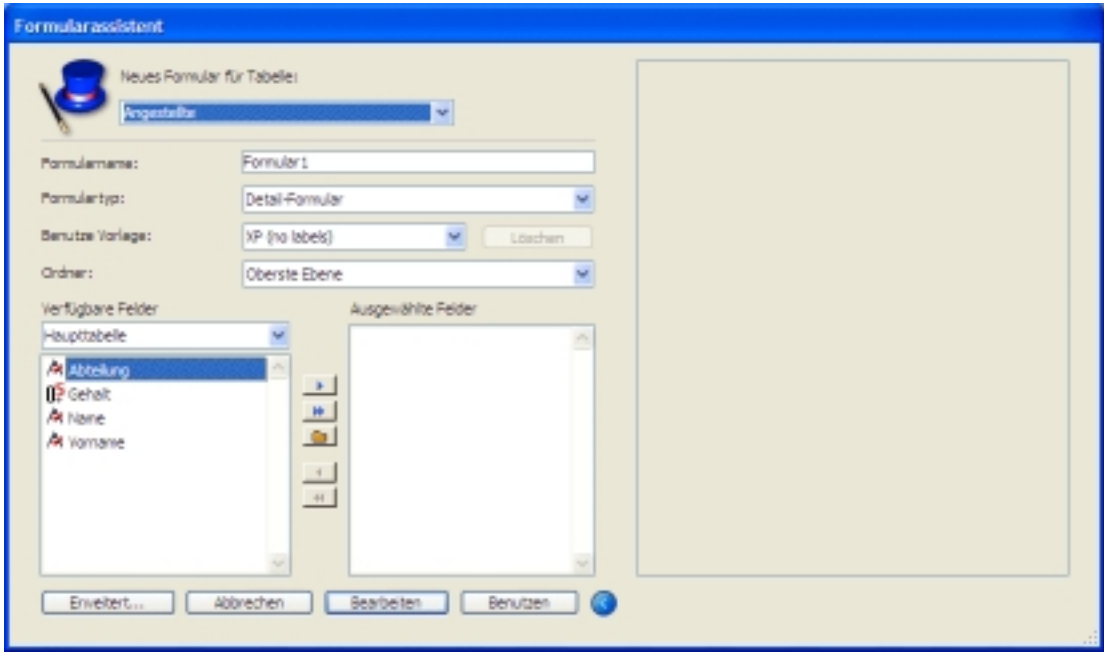
Im nächsten Abschnitt 4D Server ist ein Web Server sehen Sie, wie Sie den in 4D Server integrierten Web Server nutzen können.

Starten Sie die Server Datenbank erneut, indem Sie 4D Server starten und öffnen Sie Ihre Datenbank "Angestellte".

Melden Sie eine Arbeitsstation an der Datenbank an.

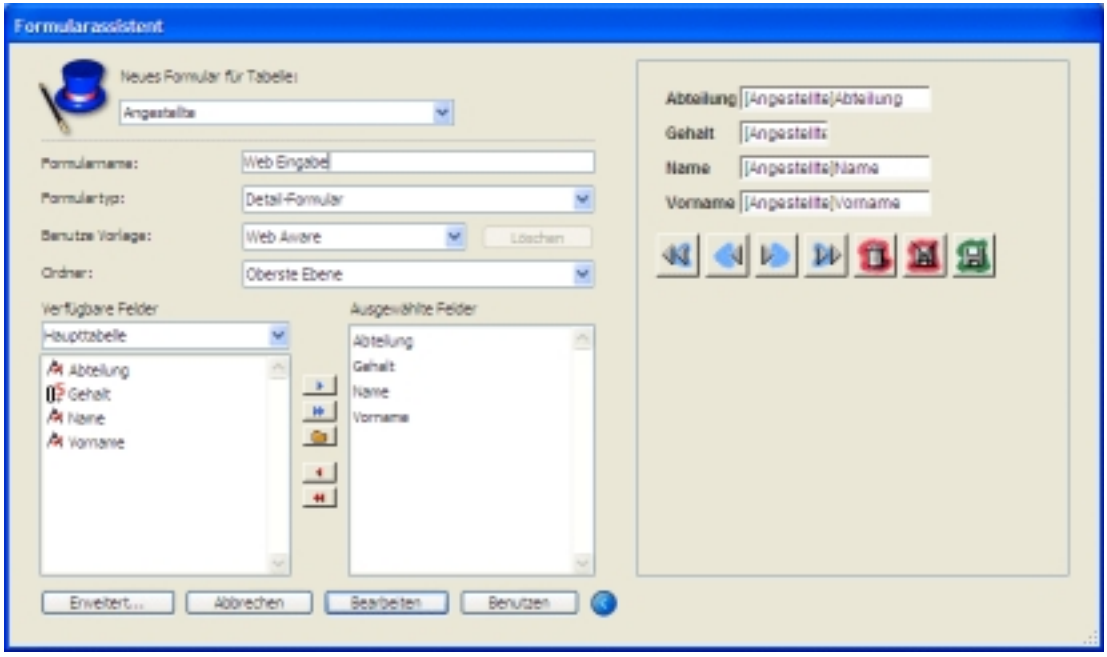
Erstellen Sie rasch ein Formular, das sich zur Veröffentlichung im Web eignet.
Gehen Sie in den Designmodus und wählen Sie im Menü **Datei/Ablage** den Befehl **Neu >Formular**.

Auf dem Bildschirm erscheint der Formularassistent:



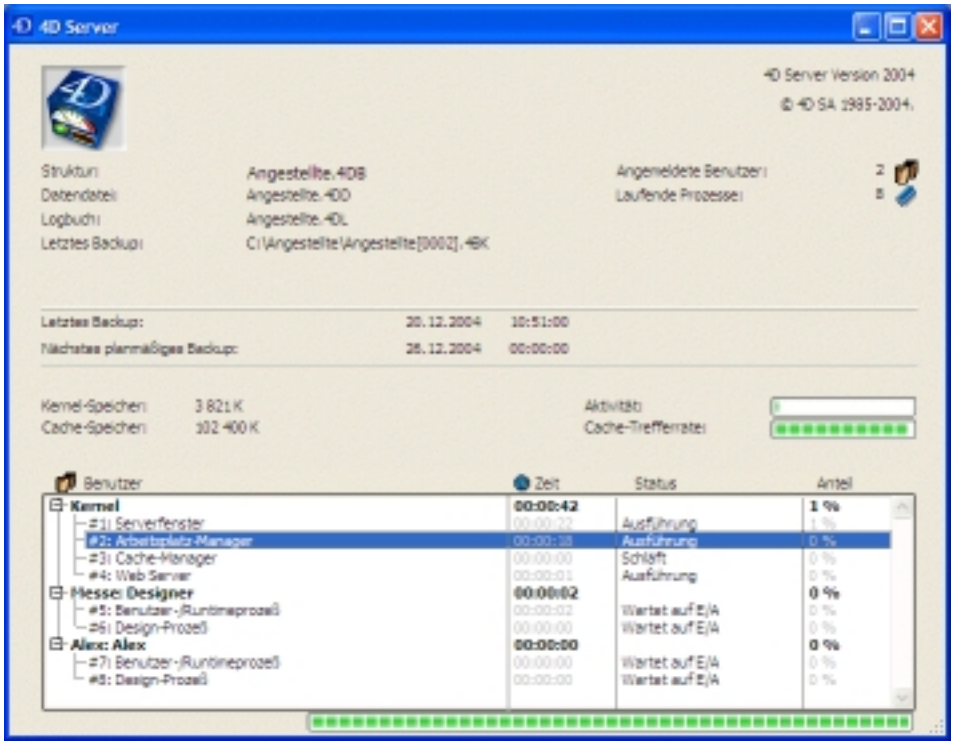
Nehmen Sie folgende Einstellungen vor:
• Formularname: “Web Eingabe”
• Formulartyp: Detail-Formular
• Verwendete Vorlage: Web Aware
• Verfügbare Felder: Klicken Sie auf die Schaltfläche “Alle Felder hinzufügen”.

Der Assistent sollte nun folgendermaßen aussehen:



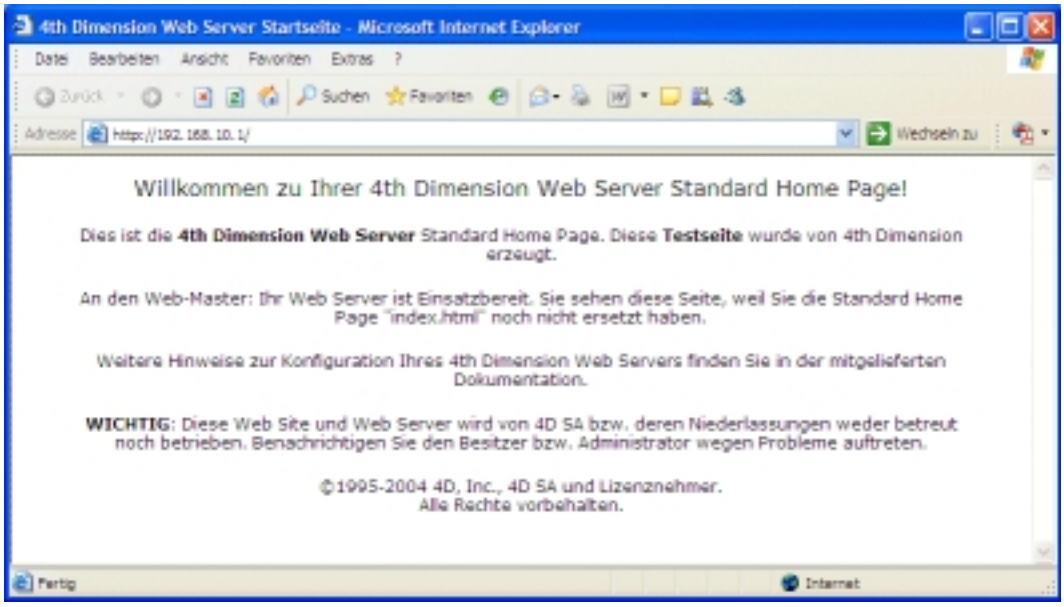
Klicken Sie auf die Schaltfläche **Benutzen**.

Auf dem Server-Rechner gibt es einen angemeldeten Benutzer:



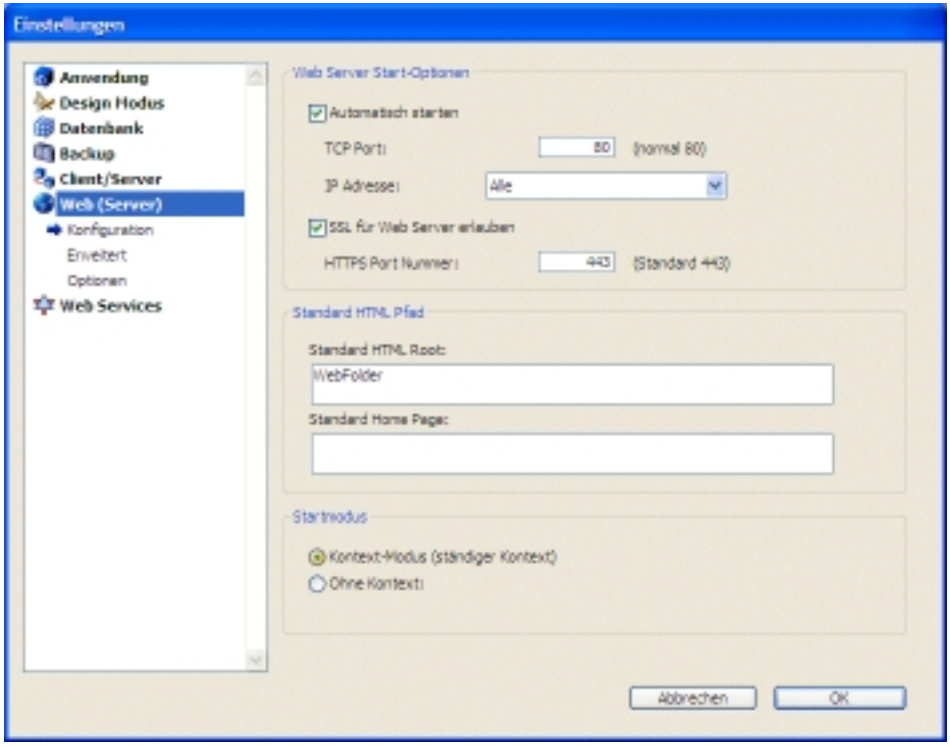
Prüfen Sie zuerst im Menü **Web Server**, ob der Web Server aktiviert ist. Wenn nicht, starten Sie ihn mit dem Menübefehl **Start Web Server**.
Um zu prüfen, ob der Web Server läuft, starten Sie einen Browser auf einem anderen Rechner und geben Sie die IP Adresse des Server-Rechners im "Adressbereich" des Browsers ein (in der Regel vom Typ 123.45.67.89).

Auf dem Bildschirm sollte die Standard Home Page des Web Servers erscheinen:


4th Dimension Web Server Startseite - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Suchen Favoriten
Adresse http://192.168.10.1/
Wechseln zu
Willkommen zu Ihrer 4th Dimension Web Server Standard Home Page!
Dies ist die 4th Dimension Web Server Standard Home Page. Diese Testseite wurde von 4th Dimension erzeugt.
An den Web-Master: Ihr Web Server ist Einsatzbereit. Sie sehen diese Seite, weil Sie die Standard Home Page "index.html" noch nicht ersetzt haben.
Weitere Hinweise zur Konfiguration Ihres 4th Dimension Web Servers finden Sie in der mitgelieferten Dokumentation.
WICHTIG: Diese Web Site und Web Server wird von 4D SA bzw. deren Niederlassungen weder betreut noch betrieben. Benachrichtigen Sie den Besitzer bzw. Administrator wegen Probleme auftreten.
©1995-2004 4D, Inc., 4D SA und Lizenznehmer.
Alle Rechte vorbehalten.
Fertig
Internet

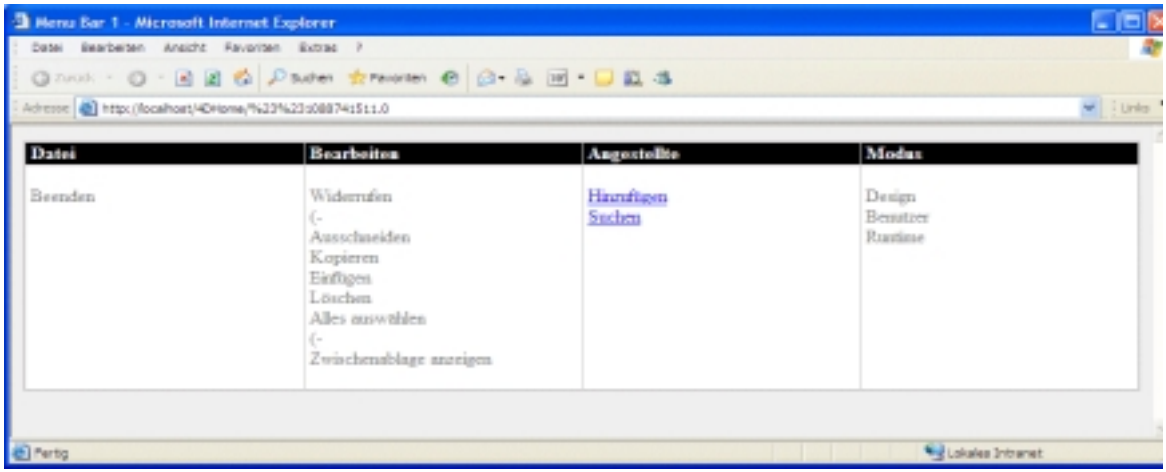
Als nächstes konfigurieren Sie den Web Server so, dass er im kontextuellen Modus startet. In diesem Modus kommen die Web Seiten von der dynamischen Umwandlung der 4D Formulare und Menüleisten. Der kontextuelle Modus eignet sich besonders zur raschen Veröffentlichung einer bestehenden Datenbank. Weitere Informationen dazu finden Sie im Handbuch 4*th Dimension Programmiersprache* im Kapitel Kontextueller Modus.
Wählen Sie auf dem Server Rechner im Menü **Bearbeiten** den Befehl **Einstellungen**, öffnen Sie die Seite “Web (Server)/Konfiguration” und führen Sie folgende Änderungen aus:
• Prüfen Sie, ob die Datenbank beim Start veröffentlicht wird
• Entfernen Sie die Standard Home Page (um die Menüleiste als Home Page zu verwenden)
• Markieren Sie die Option “Kontextueller Modus”.

Ihr Bildschirm sollte folgendermaßen aussehen:



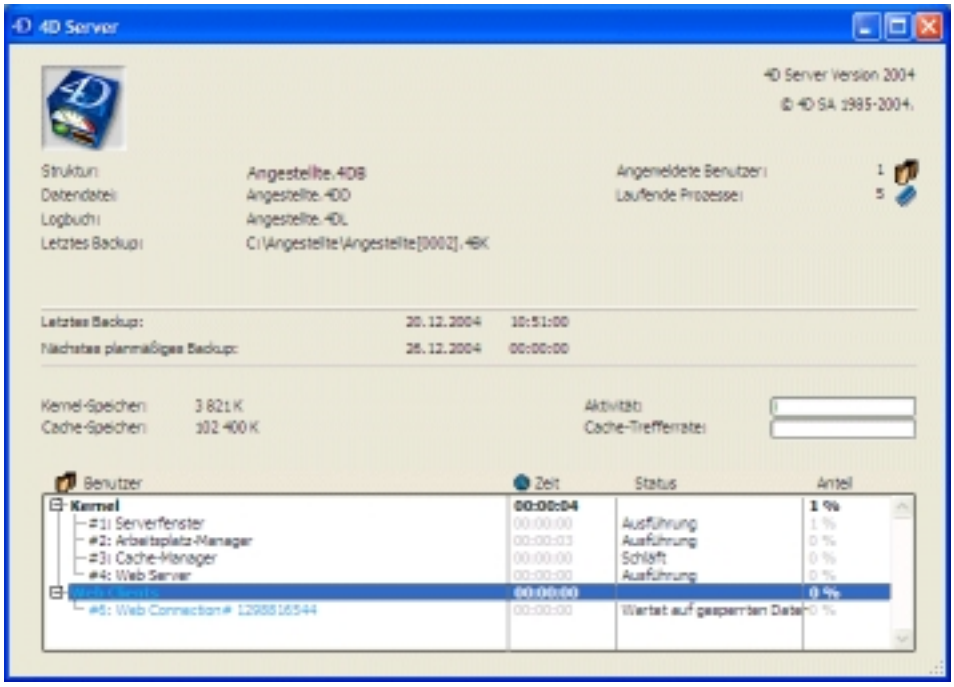
Gehen Sie zum Rechner zurück, auf welchem der Web Browser läuft und aktualisieren Sie die Server-Verbindung.

Sie erhalten nun die Web Version Ihrer eigenen Menüleiste (evtl. müssen Sie auch den Cache-Speicher des Browser leeren):



Wie Sie sehen, mussten Sie dafür weder HTML Code schreiben noch ein CGI Plug-In einbinden – 4D Server ist also auch ein Web Server!

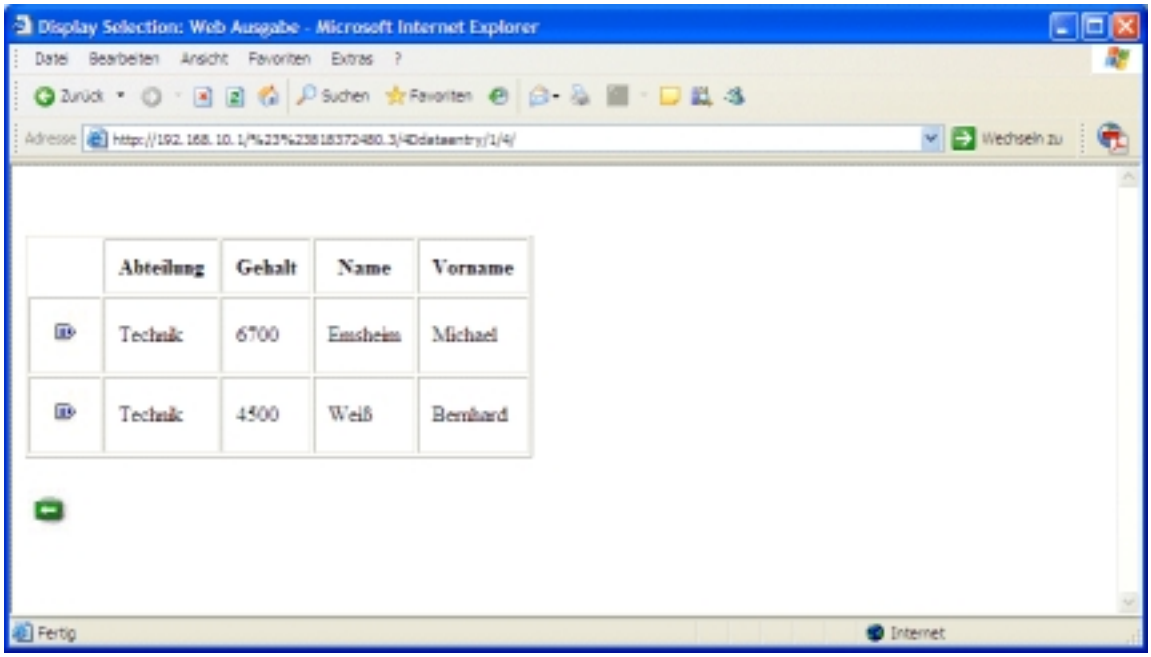
Die kontextuelle Web Anbindung erscheint auf dem Server-Rechner im Prozessfenster:



Klicken Sie im Web Browser auf die Schaltfläche **Suche**.

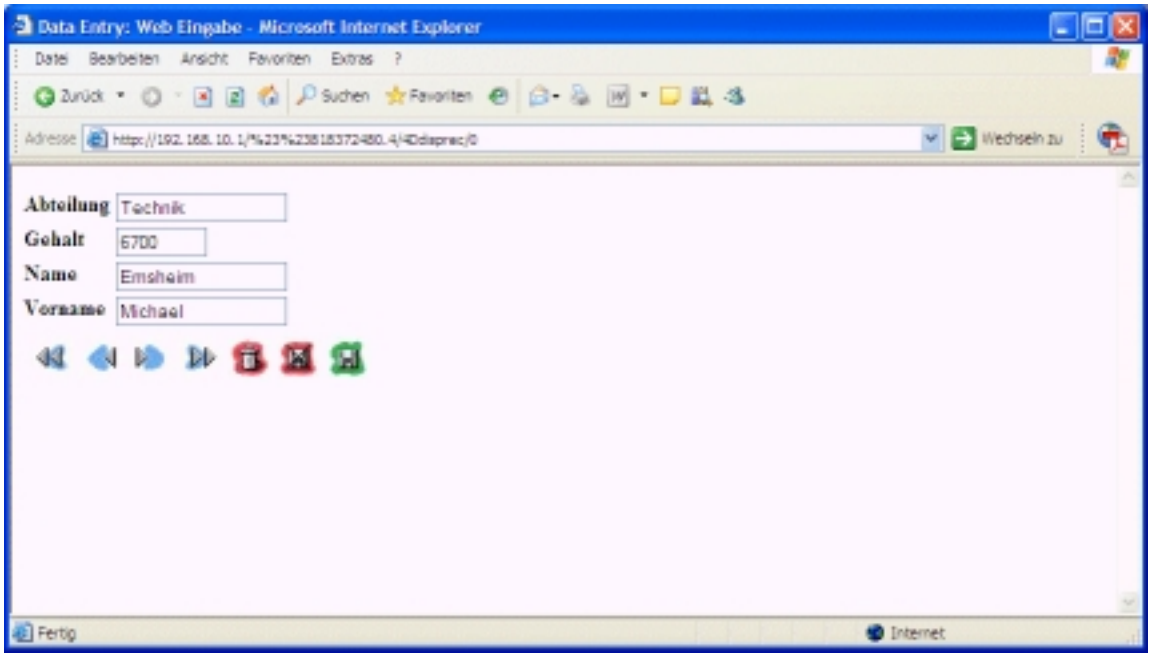
4D Server hat das Standardfenster des Sucheditors "on the fly" in eine HTML Seite übersetzt. Definieren Sie die Suche “Abteilung ist gleich Technik”.

Ihr Suchergebnis sieht folgendermaßen aus:



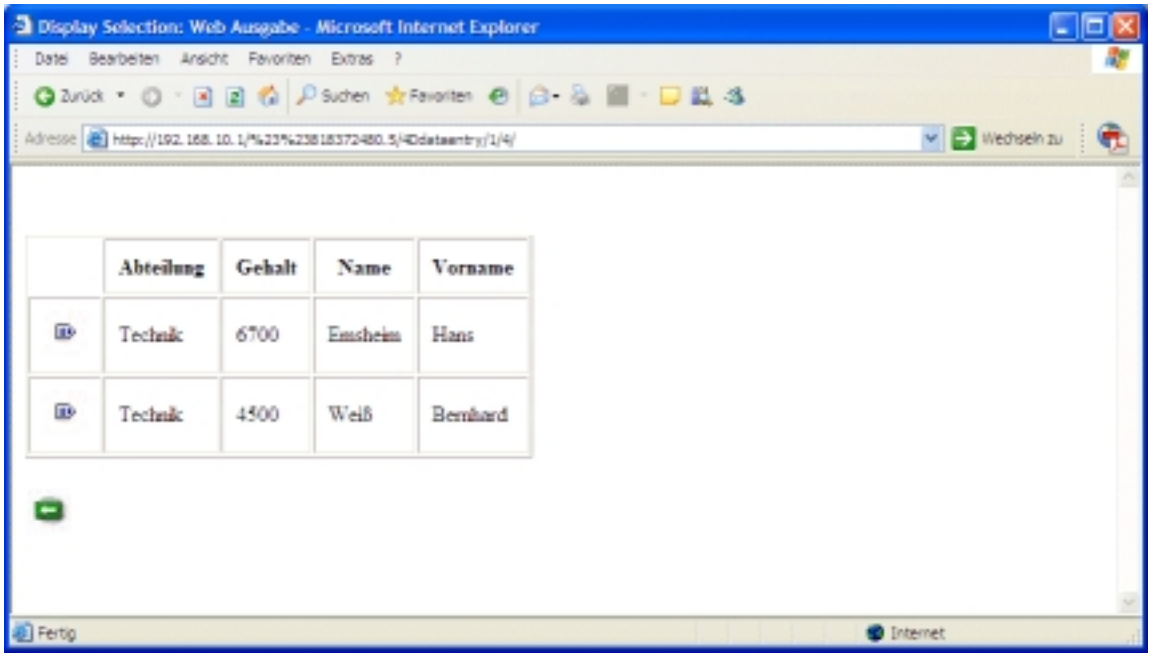
4D Server übersetzt die 4D Daten dynamisch in Web Daten und zeigt die Liste der Datensätze an.
Öffnen Sie den ersten Datensatz über das Icon in der linken Spalte.

Das Eingabeformular, das Sie mit 4D Client verwendet haben, erscheint als Web Seite:



Ändern Sie den Vornamen um in "Hans" und klicken Sie auf die Schaltfläche **Bestätigen**. Wechseln Sie in die Liste der Datensätze.

Die Änderung ist bereits berücksichtigt:



Wechseln Sie nun auf dem Client-Rechner in die Benutzerumgebung. Auch hier ist die Änderung bereits berücksichtigt:

Angestellte: 6 von 6
Abteilung Gehalt Name Vorname
Technik 6700 Emsheim Hans
Vertrieb 4500 Fürst Alexander
Versand 4000 Klein Thomas
Vertrieb 3450 Müller Andrea
Technik 4500 Weiß Bernhard
Versand 3400 Pechmann Claudia

Zusammenfassung

In nur 10 Minuten – außer Sie hatten eine Kaffeepause – haben Sie gelernt, mit 4D Server zu arbeiten:

• Sie haben von Grund auf eine Datenbank erstellt.
• Sie haben eine Tabelle eingerichtet und 4D Server angewiesen, dafür Formulare zu erstellen.
• Sie haben Datensätze angelegt und damit gearbeitet.
• Sie haben Ihre Anwendung mit einer eigenen Menüleiste ausgestattet.
• Sie haben die Server Datenbank unter Windows und auf Macintosh eingesetzt.
• Sie haben den Server abgeschaltet und erneut gestartet.
• Sie haben die Server Datenbank gleichzeitig mit 4D Client und im Web verwendet.

Schließlich haben Sie drei eigene Anwendungen erstellt (Windows, Macintosh und Web), obwohl Sie in der Tat nur eine Anwendung entwickelt haben. Außerdem können Sie die Datenbank bei Bedarf auch im Einzelplatzbetrieb einsetzen. Sie müssen die Datenbank dazu lediglich mit 4th Dimension oder 4D Runtime öffnen.

Weitere Informationen zu 4D Server finden Sie in der Einleitung und den nachfolgenden Kapiteln.

Eine vollständige Beschreibung der 4D Umgebung finden Sie in folgenden Handbüchern:

• Der *4th Dimension Designmodus* erläutert die Strukturierung von 4D Datenbanken.

• Der *4th Dimension Benutzermodus* beschreibt ausführlich die einzelnen Editoren, wie den hier erwähnten Sucheditor und den Diagrammeditor.

• Die *4th Dimension Programmiersprache* beschreibt die Befehle der 4D Programmiersprache. Wollen Sie beispielsweise mehr über 4D Server als Web-Server erfahren, lesen Sie den Abschnitt Web Server, Überblick.

3

Backup

Einleitung

4th Dimension 2004 enthält ein komplettes Modul für Backup und Wiederherstellen der Datenbank. Dieses Modul funktioniert mit 4th Dimension Einzelplatz und 4D Server. Damit können Sie von einer gerade laufenden Datenbank ein Backup erstellen, ohne sie zu beenden. Jedes Backup kann die Strukturdatei, die Datendatei sowie einzelne Segmente und zusätzliche Dateien oder Ordner enthalten. Diese Parameter setzen Sie zuerst in den Einstellungen des Programms.
Die Backups lassen sich manuell starten oder automatisch nach Zeitplan ohne Aktionen des Benutzers. Über spezifische Programmierbefehle und neue Datenbankmethoden lassen sich Backup-Funktionen in eine selbst gestaltete Oberfläche einbauen. Datenbanken werden automatisch wiederhergestellt, wenn eine beschädigte Datenbank geöffnet wird. Das integrierte Backup-Modul kann auch das Logbuch nutzen. Diese Datei schreibt alle mit den Daten ausgeführten Operationen mit und sorgt so für vollkommene Sicherheit zwischen zwei Backups. Treten bei einer laufenden Datenbank Probleme auf, werden Operationen, die in der Datendatei fehlen, beim nächsten Öffnen der Datenbank automatisch nachgetragen. Über ein Fenster können Sie den Inhalt des Logbuchs jederzeit ansehen.

Mit dem integrierten Backup-Modul können Sie:

• Jederzeit ein vollständiges Backup der Datenbankdateien starten (Strukturdatei, Datendatei, Logbuch, angehängte Dateien, etc.).

• In regelmäßigen Abständen automatische Backups einrichten – das kann stündlich, täglich, wöchentlich oder monatlich sein.

• Komplexe Backup Parameter setzen, wie Satznummer, Dateikomprimierung, Optionen für Starten nach Wiederherstellen, etc.

• Eine Datenbank mit den dazugehörige Dateien nach einem Zwischenfall automatisch wiederherstellen.

• Operationen rückgängig machen, die in den Daten der Datenbank ausgeführt wurden.

Backups verwalten

Hauptfunktion des Moduls ist das Backup. Bei einem Backup der Datenbank wird zu einem gegebenen Zeitpunkt eine Kopie der Datenbank und aller notwendigen Dateien erstellt. Diese Kopie wird in eine Backup-Datei gelegt.
Backup-Dateien lassen sich später mit 4th Dimension öffnen, wenn ein Zwischenfall eintritt, der die aktuelle Datenbank beschädigt. Die Datenbank wird dann im Status der Kopie wiederhergestellt.

Backup ausführen

Jedes Backup berücksichtigt die Parameter in den Einstellungen des Programms. In den Einstellungen der Datenbank legen Sie alle Optionen für das Backup fest:

• Die Dateien für das Backup (Daten/Logbuch, Struktur, Benutzerstruktur und dazugehörige Dateien),

• Die Platzierung der Backup-Dateien (primäre Backup-Datei und Logbuch der Backup-Datei).

• Verwaltung des Logbuchs

• Backup-Planer

• Backup Optionen: Anzahl und Frequenz der Backup-Sätze, Steuern von Transaktionen oder aktiven Indizierungen, Vorgehensweise bei Fehlschlägen, Segmentation, Komprimierung und Integrität der Backup-Dateien.

• Optionen für automatisches Wiederherstellen: Diese Parameter sind auf Standardwerte für normalen Einsatz gesetzt; Sie können sie bei Bedarf verändern. Weitere Informationen dazu finden Sie im Absatz “Backup Einstellungen”.

Backup starten

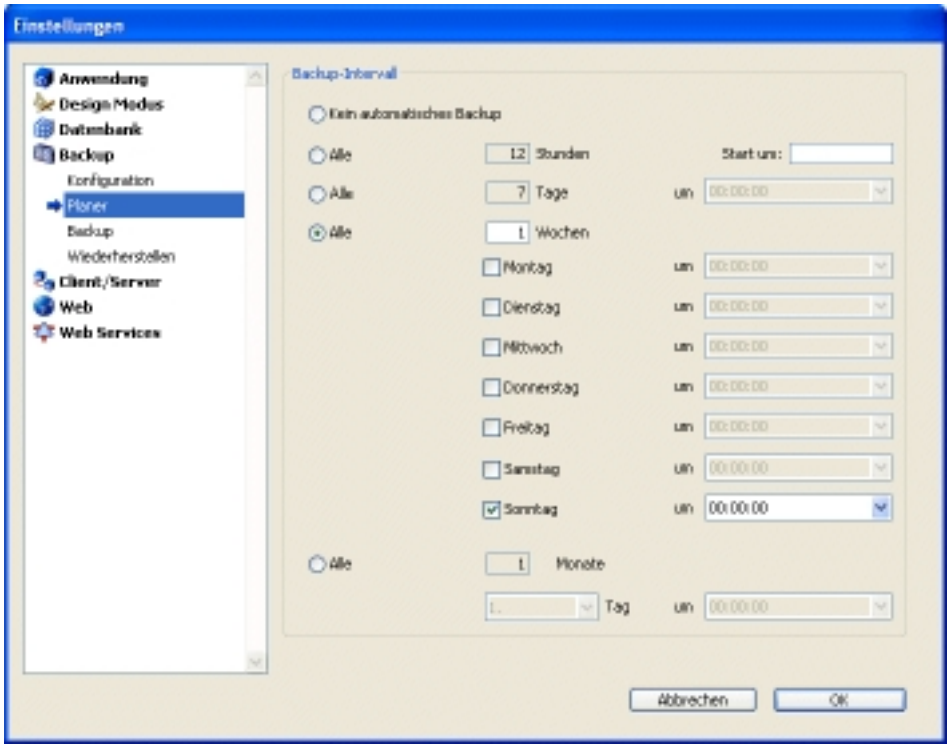
In 4th Dimension 2004 gibt es drei Möglichkeiten, ein Backup zu starten:

• Manuell über den Befehl Backup im Menü **Datei/Ablage** von 4th Dimension (Benutzerumgebung) bzw. von 4D Server.

• Automatisch über den Planer, der in den Einstellungen des Programms aktiviert wird.

• Per Programmierung über den Befehl BACKUP. Ein Backup lässt sich über eine Methode, die den Befehl BACKUP aufruft, auch manuell vom 4D Client aus starten. In allen Fällen wird er auf dem Server ausgeführt.

• Um ein Backup automatisch nach Plan auszuführen, gehen Sie in den Einstellungen der Datenbank zum Thema “Backup” und dann auf die Seite Planer:

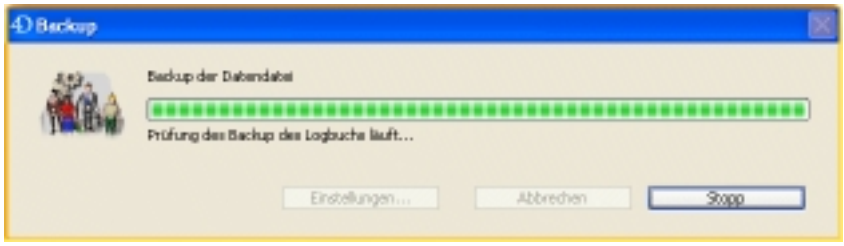


Das Backup wird automatisch im hier definierten Intervall ausgeführt, ohne dass der Benutzer irgendetwas tun muss. Weitere Information dazu finden Sie im Absatz “Backup nach Zeitplan”.

• Um ein Backup nach Plan über Programmierung auszuführen, führen Sie den Befehl BACKUP in einer Methode aus. Das Backup startet mit den aktuellen Parametern.
Zur Abwicklung des Backups können Sie auch die neuen Datenbankmethoden Backup Startup und On Backup Shutdown verwenden. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* im Kapitel Backup.

Backup ausführen

Sobald das Backup gestartet ist, zeigt 4th Dimension das Fortschreiten der Operation in einem Dialogfenster an:



Über die Schaltfläche **Stopp** kann der Benutzer das Backup jederzeit unterbrechen. Weitere Informationen dazu finden Sie im nächsten Absatz.

Das Ergebnis des letzten Backup (erfolgreich oder fehlgeschlagen) wird in den Einstellungen von **Backup** auf der Seite **Konfiguration** im Bereich “Letzte Backup Information” gespeichert oder im Hauptfenster von 4D Server. Weitere Informationen dazu finden Sie im Absatz “Letztes Backup". Es wird ausserdem im Backup-Journal der Datenbank abgelegt. Weitere Informationen dazu finden Sie im Abschnitt Backup Journal.

Während Backup auf die Datenbank zugreifen

Während ein Backup läuft, können Sie folgendermaßen auf die Datenbank zugreifen:

• In 4th Dimension Einzelplatz ist die Datenbank im Lese- und Schreibmodus gesperrt; alle Prozesse sind eingefroren. Es lassen sich keine Aktionen ausführen.

• In 4D Server ist die Datenbank nur im Schreibmodus gesperrt; Client-Rechner können die Daten ansehen. Sendet ein Client eine Anfrage für Hinzufügen, Andern oder Entfernen an den Server, erscheint eine Meldung, dass der Benutzer bis zum Ende des Backup warten muss.
Ist die Datenbank gesichert, verschwindet das Fenster und die Aktion wird ausgeführt. Um die laufende Anfrage abzubrechen und nicht das Ende des Backup abzuwarten, klicken Sie auf die Schaltfläche **Abbrechen**.

- Stammt die Aktion in Warteposition jedoch von einer Methode, die vor dem Backup gestartet wurde, sollte sie nicht abgebrochen werden, da nur noch verbleibende offene Operationen abgebrochen werden. Darüberhinaus kann eine nur teilweise ausgeführte Methode zu Inkonsistenzen in der Datenbank führen.

- Stammt die Aktion in Warteposition von einer Methode und klickt der Benutzer auf die Schaltfläche **Abbrechen**, gibt 4D Server Fehler -9976 zurück (Befehl kann nicht ausgeführt werden, da Backup läuft).

Probleme während Backup-Ausführung

Es kann passieren, dass ein Backup nicht korrekt ausgeführt wird. Das kann verschiedene Ursachen haben: Unterbrechung durch den Benutzer, angefügte Datei wird nicht gefunden, Probleme auf dem Zieldatenträger, unvollständige Transaktion, etc. 4th Dimension bearbeitet den Zwischenfall je nach Ursache.
In allen Fällen wird der Status des letzten Backup (erfolgreich oder fehlgeschlagen) angezeigt, und zwar in den Einstellungen von Backup auf der Seite **Konfiguration**, im Fenster 4D Server und im Backup-Logbuch. Weitere Informationen dazu finden Sie im Absatz “Letztes Backup”.

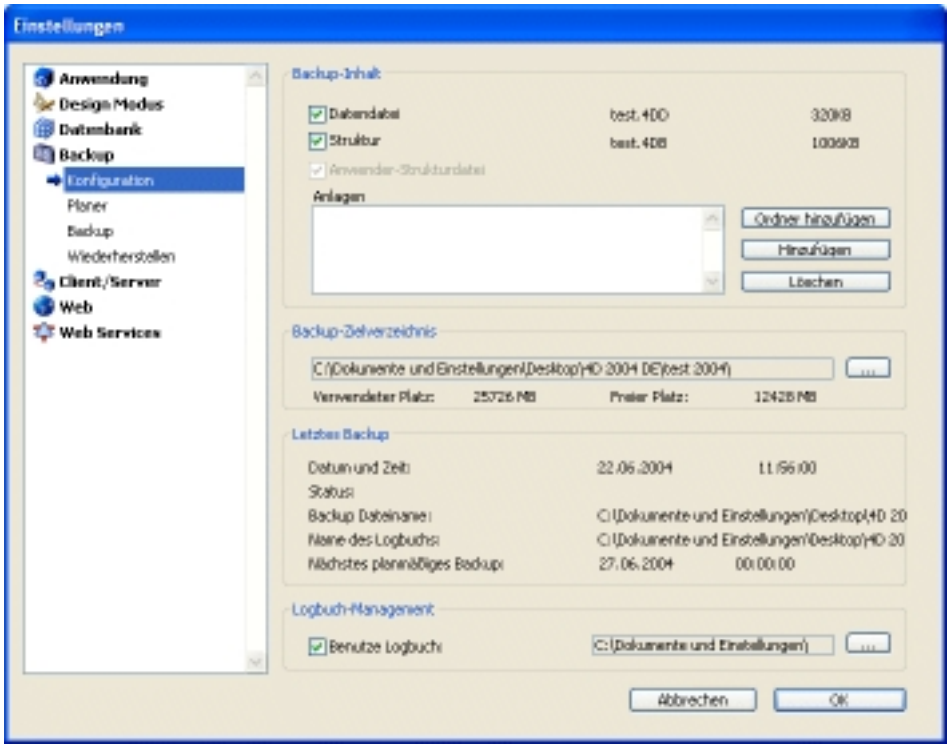
• **Unterbrechung durch den Benutzer:** Der Benutzer kann das Backup im Dialogfenster, das den Verlauf anzeigt, jederzeit über die Schaltfläche **Stopp** abbrechen. Dann stoppt das Kopieren von Elementen und ein Fehler wird angezeigt. Sie können diesen Fehler über die Datenbankmethode Backup Shutdown abfangen.

• **Angefügte Datei wird nicht gefunden:** Ist eine angefügte Datei nicht auffindbar, führt 4th Dimension das Backup teilweise durch, d.h. Backup der Dateien der Datenbank und verfügbaren angefügten Dateien, und gibt einen Fehler zurück.

• **Backup ist nicht möglich:** (Platte ist voll oder schreibgeschützt, Platte fehlt oder ist fehlerhaft, unvollständige Transaktion, Datenbank zum Zeitpunkt für geplantes automatischen Backup nicht gestartet, etc.)
- Tritt der Fehler zum ersten Mal auf, versucht 4th Dimension erneut, das Backup auszuführen. Die Wartezeit zwischen zwei Versuchen wird in den Einstellungen des Backup auf der Seite **Konfiguration** ausgeführt. Weitere Informationen dazu finden Sie im Absatz “Wenn Backup fehlschlägt”.
- Scheitert auch der zweite Versuch, erscheint ein Meldedialog des Systems und ein Fehler wird angezeigt. Sie können diesen Fehler über die Datenbankmethode On Backup Shutdown abfangen.
- Backups, die aufgrund einer unvollständigen Transaktion bzw. Indizierung abgeblockt werden, werden über einen spezifischen Mechanismus verwaltet. Weitere Informationen dazu finden Sie im Absatz “Wenn Transaktionen oder Indizierungen aktiv”.

Backup-Dateien konfigurieren

In den Einstellungen der Datenbank unter dem Thema Backup auf der Seite **Konfiguration** bestimmen Sie die Backup-Dateien, das Logbuch und ihre Position. Hier erhalten Sie auch Informationen zum letzten Backup.



Diese Parameter sind nur auf dem Rechner mit 4D Server zugänglich.
Die gezeigten Parameter sind für jede mit 4th Dimension geöffnete Datenbank anders.

Backup Inhalt

Hier definieren Sie, welche Dateien bzw. Ordner beim nächsten Backup kopiert werden. Der obere Bereich zeigt die Dateien der 4th Dimension Datenbank und jeweils ihre aktuelle Größe an. Markieren Sie die Dateien, welche das Backup enthalten soll. Eine Option in Grauschrift gibt an, dass die Datei in der Datenbank nicht vorhanden ist. Sie können die Dateien beliebig auswählen. Die Wahl richtet sich nach Kriterien wie Häufigkeit der Aktualisierung, strategische Bedeutung, Größe, o.ä.. Keine Datei ist zwingend erforderlich.
Der darunterliegende Bereich „Anlagen“ listet den Zugriffspfad für angefügte Dateien auf, die das Backup berücksichtigen soll.

• **Datendatei:** Besteht die Datenbank aus mehreren Segmenten, erscheinen nur Name und Größe des ersten Segments. Es wird natürlich von allen Segmenten ein Backup erstellt.

Mit dieser Option wird auch – sofern vorhanden – zeitgleich vom aktuellen Logbuch der Datenbank ein Backup erstellt. Dabei wird das aktuelle Logbuch geschlossen und als Backup gesichert und anschließend ein neues Logbuch angelegt. Diese Operation verhindert, dass das Logbuch zu groß wird. Weitere Informationen dazu finden Sie im Absatz Logbuch verwalten.

• **Strukturdatei:** Diese Option ermöglicht bei Datenbanken mit einkompilierter 4D Runtime Volume Licence unter Windows ein Backup der .exe Datei, auf MacOS des Software-Pakets.

• **Anwender-Strukturdatei:** Diese Datei enthält angepasste Benutzerformulare. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* im Kapitel “Formulare".

• **Anlagen:** Hier können Sie Dateien bzw. Ordner wählen, die zusammen mit der Datenbank als Backup gesichert werden sollen. Das können Dokumente, Vorlagen von Plug-Ins, Etiketten, Berichte, Bilder, o.ä. sein.
Sie können eigene Dateien oder Ordner bestimmen, deren Inhalt vollkommen als Backup gesichert wird. Jedes angefügte Element wird mit dem kompletten Zugriffspfad im Bereich „Anlagen“ aufgeführt.
- **Ordner hinzufügen:** Klicken Sie auf diese Schaltfläche, öffnet 4th Dimension ein Dialogfenster, um Ordner für das Backup auszuwählen. Beim Wiederherstellen wird dieser Ordner mit seiner internen Struktur wiederhergestellt. Sie können jeden Ordner bzw. jedes Verzeichnis wählen, das an den Rechner angeschlossen ist. Die Ordner mit den Dateien der Datenbank können Sie nicht als Anlage hinzufügen.
- **Hinzufügen:** Klicken Sie auf diese Schaltfläche, öffnet 4th Dimension ein Dialogfenster, um Dateien für das Backup auszuwählen. Eine Datei der Datenbank können Sie nicht als Anlage hinzufügen.
- **Löschen:** Klicken Sie auf diese Schaltfläche, wird die gewählte Datei aus der Liste entfernt.

Hinweis: Unter Umständen sind ein bzw. mehrere angefügte Dateien nicht zugänglich, wenn das Backup ausgeführt wird, z.B. weil sich Name oder Zugriffspfad geändert haben oder das Verzeichnis voll ist. In diesem Fall wird das Backup ohne die fehlenden Dateien ausgeführt und ein Fehler gemeldet. Sie können den Fehler mit der Datenbankmethode On Backup Shutdown abfangen. Der Fehler erscheint auch im Logbuch des Backup.

Backup-Zielverzeichnis

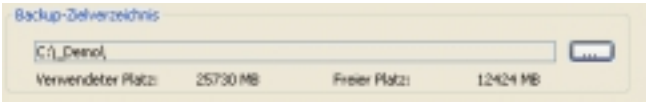
Hier definieren Sie, wo die Backup-Dateien und Logbücher gespeichert werden sollen. 4th Dimension erstellt zwei Arten von Backup-Dokumenten: Backup-Dateien und Backup-Logbücher. Dateien haben folgende spezifischen Merkmale:

• Sie können mehrere Dateien enthalten, wie Struktur-, Daten-, und angefügte Dateien.

• Sie sind durch einen internen Überprüfungsmechanismus abgesichert. Dafür gibt es verschiedene Möglichkeiten in den erweiterten Einstellungen für Backup.

Hinweis: 4th Dimension speichert diese Dateien standardmäßig neben der Datendatei der Datenbank. Wir raten jedoch dringend, die Dateien auf einer anderen Festplatte abzulegen, um zu vermeiden, dass bei Ausfall der Platte, die Datenbank und Backup enthält, sämtliche Daten verloren gehen.

Um den Speicherort dieser Dateien zu ändern, klicken Sie auf die Schaltfläche [...]:



Auf dem Bildschirm erscheint ein Auswahldialog, in dem Sie einen Ordner bzw. eine Festplatte zum Ablegen der Backups wählen können. Die Bereiche “Verwendeter Platz” und “Freier Platz” werden automatisch aktualisiert. Sie geben an, wieviel Speicherplatz auf der Festplatte mit dem gewählten Ordner belegt und wieviel frei ist.
Stellen Sie sicher, dass der freie Platz für all Ihre Backups ausreicht. Schlägt ein Backup fehl, weil der Platz nicht ausreicht, wird ein Fehler erzeugt. Sie können diesen Fehler in der Datenbankmethode On Backup Shutdown abfangen. Der Fehler erscheint auch im Informationsbereich und im Logbuch des Backup.

Namen der Backup Dateien

Backup-Dateien in 4th Dimension verwenden ein spezifisches Namenssystem, auf dem die automatischen Funktionen zur Wiederherstellung basieren. Dieses Namenssystem lässt sich nicht verändern.

• Standard Backups lauten **Datenbankname[xxxx].4BK**, wobei Datenbankname der Name der Datendatei der Datenbank ist und xxxx die Nummer des Backup. Beispiel: Das 26. Backup der Datenbank Rechnungen hat den Namen Rechnungen[0026].4BK. Ist das Backup in Segmente unterteilt, fügt 4th Dimension die Segmentnummer als -xxxx hinzu. Beispiel: Das 3. Segment des 26. Backup der Datenbank Rechnungen hat den Namen Rechnungen[0026-0003].4BK.
Weitere Informationen zu Segmenten finden Sie im Absatz Archiv.

• Backups von Logbüchern lauten **Logname[xxxx].4BL**, wobei Logname der Name des Logbuchs „Log“ der Datenbank ist und xxxx die Nummer des Backup (beginnt mit 0).

Beispiel: Das 13. Backup des Logbuchs „Log“ hat den Namen Log[0012].4BL.
Ist das Backup des Logbuchs in Segmente unterteilt, fügt 4th Dimension die Segmentnummer als -xxxx hinzu.

Beispiel: Das 2. Segment des 13. Backup des Logbuchs „Log“ hat den Namen Log[0012-0002].4BL.

Beachten Sie, dass Backups von Logbüchern bei 0 (Null) starten, Backups von Datenbankdateien dagegen bei 1. Lautet z.B. das erste Backup der Datenbank Backup[0001].4BK, hat das dazugehörige Logbuch den Namen Log[0000]: Es enthält die Anderungen in der Datendatei, beginnend mit dem Status “leer” und kann nur in eine leere Datendatei integriert werden. Folglich muss ein Logbuch mit Namen Log[0025].4BL als das 26. Backup interpretiert werden, das die Operationen enthält, die zwischen dem 25. und 26. Backup der Datenbank ausgeführt wurden. Log[0025].4BL gehört also zum Backup Backup[0025].4BK.

Die Backup Nummerierung lautet:
• Backup: 1 bis 9999
• Backup des Logbuchs: 0 bis 9998
• Segment: 1 bis 9999

Letztes Backup: Dieser Bereich zeigt Informationen zum letzten Backup der Datenbank. Dazu muss mindestens ein Backup ausgeführt worden sein.

• **Datum und Zeit:** Datum und Uhrzeit des letzten Backup.

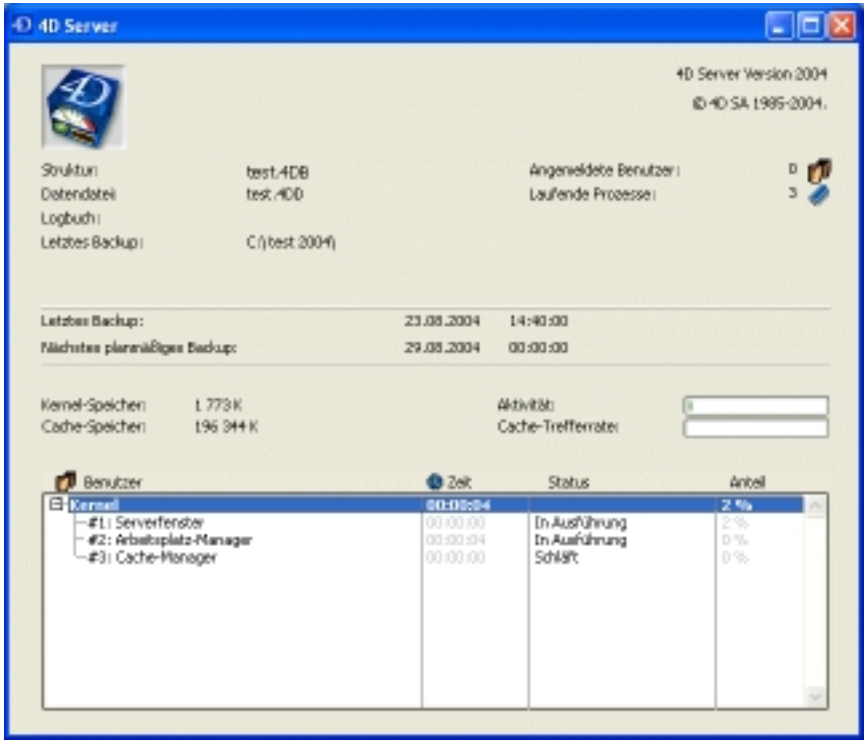
• **Status:** Hier erscheint die Fehlernummer mit der dazugehörigen Meldung für das letzte Backup. Bei korrekter Ausführung ist der Status 0 (Null), d.h. das Backup war erfolgreich.

• **Backup Dateiname:** Zugriffspfad und Dateiname des letzten Haupt-Backup.
Besteht das Backup aus mehreren Segmenten, erscheint der Name des ersten Segments.

• **Name des Logbuchs:** Zugriffspfad und Dateiname des letzten Logbuchs des Backups – sofern zutreffend.

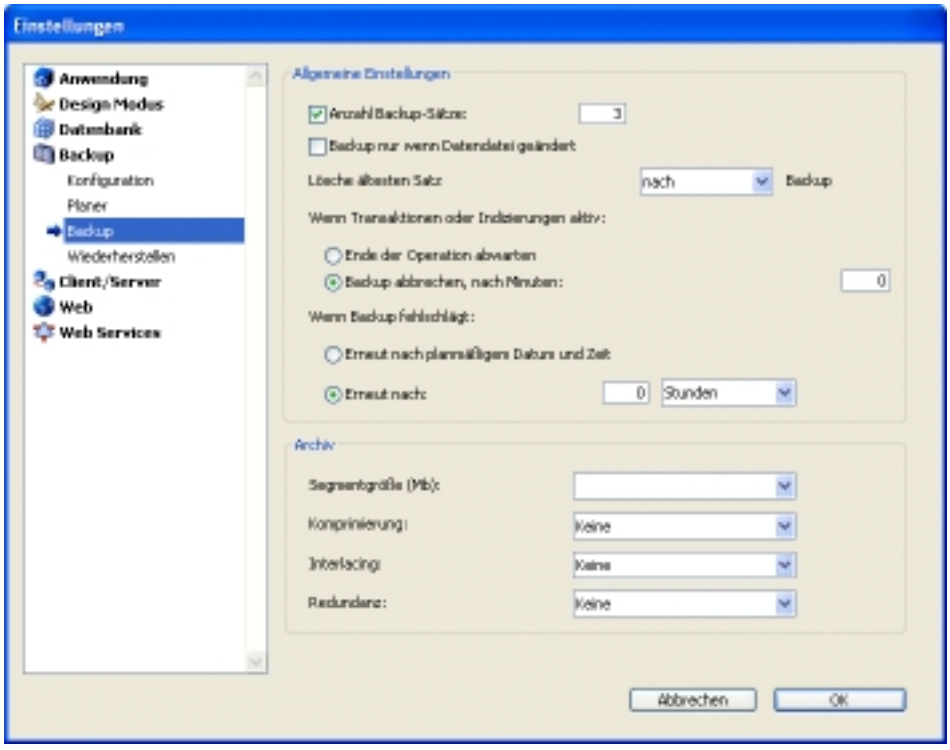
• **Nächstes planmäßiges Backup:** Datum und Uhrzeit des nächsten Backup; Hier erscheinen Informationen, wenn ein Zeitplan für das Backup eingerichtet wurde.

Auf 4D Server erscheint diese Information auch im Server-Hauptfenster:



Backup Einstellungen

Die Backup-Einstellungen gelten wie die Einstellungen zur Konfiguration ebenfalls für jedes Backup. Die eingetragenen Werte entsprechen der Standardverwendung. Sie können die Werte bei Bedarf verändern. Dies definieren Sie in den Einstellungen der Datenbank auf der Seite Backup:



Im Client-/Server-Betrieb lassen sich die Backup-Einstellungen nur auf dem Rechner mit 4D Server setzen. Die Einstellungen gelten für jede Datenbank, die mit dem Programm 4th Dimension geöffnet werden.

Anzahl Backup-Sätze

Damit legen Sie fest, dass die älteren Backup-Dateien gelöscht werden. Im Eingabebereich definieren Sie, wieviel Backups Sie aufbewahren wollen. Wird die Zahl überschritten, wird das älteste Backup beim nächsten Backup gelöscht. So vermeiden Sie eine Überbelegung der Festplatte.
Das funktioniert folgendermassen: Ist das aktuelle Backup komplett, löscht 4th Dimension das älteste Archiv, wenn es an derselben Stelle liegt und denselben Namen hat. Sie können auch bestimmen, dass das älteste Archiv vor Ausführen des Backup gelöscht wird, um Platz auf der Festplatte zu sparen.
Ist z.B. als Anzahl 3 definiert, erstellen die ersten drei Backups jeweils die Archive MeineDB-0001, MeineDB-0002 und MeineDB-0003. Beim 4. Backup wird das Archiv MeineDB-0004 erstellt und MeineDB-0001 gelöscht.

Ausgehend vom Platz auf der Festplatte, wo Sie Ihre Backups ablegen, können Sie die max. Anzahl der Backup-Sätze mit folgender Gleichung festlegen:

$$\text{Max. Anzahl der Sätze} = \frac{\text{Verfügbarer Platz für Backups}}{\text{Max. Größe des Backup}} - 1 \text{ Satz}$$

Sie müssen die max. Anzahl der Sätze um 1 reduzieren, da 4th Dimension standardmäßig zuerst das aktuelle Backup durchführt, und anschließend das älteste Archiv auf der Festplatte löscht. Dieses Verhalten können Sie ändern über die Option “Backup nur wenn Datendatei geändert”. Nähere Informationen dazu finden Sie im nächsten Absatz. Standardmäßig ist der Löschmechanismus aktiviert und 4th Dimension behält 3 Backup-Sätze.
Wollen Sie diesen Mechanismus deaktivieren, deaktivieren Sie einfach diese Option.

Hinweis: Dieser Parameter betrifft sowohl die Backup-Sätze der Datenbank als auch das Logbuch der Backup-Sätze.

Backup nur wenn Datendatei geändert

Ist diese Option markiert, startet 4th Dimension geplante Backups nur, wenn seit dem letzten Backup Daten in der Datenbank hinzugefügt, verändert oder gelöscht wurden. Sonst wird das geplante Backup nicht ausgeführt und auf das nächste geplante Backup gelegt. Es erscheint keine Fehlermeldung; das Logbuch notiert jedoch, dass das Backup verschoben wurde.
Damit sparen Sie Rechnerzeit für Backups von Datenbanken, die überwiegend zum Ansehen genutzt werden. Beachten Sie jedoch, dass mit dieser Option keine Anderungen an der Struktur oder angefügten Dateien berücksichtigt werden.

- Weitere Informationen dazu finden Sie im Absatz “Backup nach Zeitplan”.
- Ist diese Option markiert, und startet das Backup manuell, erscheint die Meldung, dass die Datendatei nicht geändert wurde. Der Benutzer kann die Operation bestätigen oder abbrechen.
- Dieser Parameter betrifft sowohl Backups der Datenbank als auch des Logbuchs.

Lösche ältesten Satz nach/vor Backup

Diese Option wird nur verwendet, wenn die Option “Anzahl der Backups” markiert ist. Damit geben Sie an, ob 4th Dimension das älteste Archiv vor (Option **vor**) oder nach Ausführen des Backup (Option **nach**) löscht. Das funktioniert nur, wenn das älteste Archiv weder umbenannt noch verschoben wurde.

Mit **vor** sparen Sie Speicherplatz. Es ist nicht notwendig, die Anzahl der Backups um 1 zu verringern. Dadurch lassen sich z.B. zwei Archive mit je 2 GB auf einer Festplatte mit 5 GB speichern. Beachten Sie, dass Sie bei Unterbrechung eines Backups weder auf das alte (das ja zuvor gelöscht wurde), noch auf das aktuelle Backup (das noch nicht abgeschlossen ist) zugreifen können. Da sich die Datenbank jedoch während dem Backup nicht verändern lässt, können Sie sicher sein, dass die Datendatei intakt ist, selbst wenn Sie kein gültiges Backup haben. Startet das System neu und ist das unvollständige Archiv zerstört, sollten Sie gleich ein neues Backup erstellen.

Die Option **nach** ist eine zusätzliche Sicherheitsmaßnahme, die jedoch mehr Platz auf der Festplatte mit dem Backup benötigt.

Wenn Transaktionen oder Indizierungen aktiv

Transaktionen und Indizierungen sind kritische Operationen, die kein Backup der Datenbank zulassen, solange sie nicht abgeschlossen sind. Vor Starten eines Backup – egal auf welche Weise – prüft 4th Dimension oder 4D Server die Datenbank; stösst das Programm auf eine gerade laufende kritische Operation, wird der Backup-Prozess auf Warteposition gesetzt.
Dafür können Sie folgende Parameter festlegen:

• **Ende der Operation abwarten:** Das Backup ist in Wartestellung, solange die Operation nicht abgeschlossen ist; und startet unmittelbar danach. So ist garantiert, dass das Backup ausgeführt wird. Es lässt sich jedoch nicht genau überprüfen, wann das Backup stattfindet.

• **Backup abbrechen nach Minuten:** Das Backup wartet eine bestimmte Zeitspanne ab, damit die Transaktion bzw. Indizierung beendet werden kann. Standardmäßig sind 3 Minuten eingestellt. Ist die Operation nach dieser Zeit noch nicht abgeschlossen, gilt das Backup als fehlgeschlagen. Im Logbuch des Backup wird ein Fehler notiert. Dann läuft der Vorgang ab, der für fehlgeschlagene Backups definiert wurde. (siehe folgenden Absatz).

Hinweise: Vermeiden Sie das Öffnen einer Transaktion beim Starten einer Datenbank sowie eine Transaktion, die nur beim Beenden der Datenbank bestätigt bzw. abgebrochen wird. Sie sollten auch während einer Transaktion keine Dialogfenster einrichten, in die der Benutzer etwas eintragen oder die er bestätigen muss. Startet das Backup in diesem Moment, kann die Datenbank neue Transaktionen erst ausführen, wenn das Dialogfenster bestätigt wurde.

Wenn Backup fehlschlägt

Damit legen Sie fest, wie fehlgeschlagene Backups verwaltet werden.

Hinweise:

- Nicht alle Zwischenfälle führen zu einem fehlgeschlagenen Backup. Weitere Informationen dazu finden Sie im Absatz “Probleme während Backup-Ausführung”.
- 4th Dimension wertet ein Backup als fehlgeschlagen, wenn die Datenbank zum Zeitpunkt des automatisch geplanten Backups nicht gestartet ist. Weitere Informationen dazu finden Sie im Absatz “Backup nach Zeitplan”.

Lässt sich ein Backup nicht ausführen, ermöglicht 4th Dimension einen zweiten Versuch. Dafür gibt es zwei Optionen:

• **Erneut nach planmäßigem Datum und Zeit:** Diese Option ist nur sinnvoll für geplante automatische Backups. Weitere Informationen dazu finden Sie im Absatz “Backup nach Zeitplan”. Das fehlgeschlagene Backup wird abgebrochen und ein Fehler erscheint.

• **Erneut nach X Stunden oder Minuten:** Mit dieser Option wird nach der Wartezeit ein neues Backup versucht. So lassen sich bestimmte Umstände umgehen, die ein Backup blockieren können. Sie können die Wartezeit in Stunden oder Minuten angeben.

Scheitert auch der 2. Versuch, erscheint ein Fehler. Das wird auch im Statusbereich des letzten Backup und im Logbuch des Backup vermerkt.

Archiv

Hier legen Sie die Einstellungen zum Anlegen des Archivs fest. Sie gelten für die Dateien von Haupt-Backups sowie für Logbücher der Backups.

• **Segmentgröße (MB):** Mit 4th Dimension können Sie Archive segmentieren, d.h. ein Archiv in mehrere kleinere aufteilen. So können Sie z.B. ein Backup über verschiedene Medien, wie CD, ZIP o.ä. speichern. Beim Wiederherstellen fügt 4th Dimension die Segmente automatisch wieder zusammen. Jedes Segment hat den Namen MeineDBxxxx-yyyy].4BK, wobei xxxx die Nummer des Backup ist, yyyy die Nummer des Segments. Beispiel: Die drei Segmente der Datenbank MeineDB lauten MeineDB[0006-0001].4BK, MeineDB[0006-0002].4BK und MeineDB[0006-0003].4BK.
Segmentgröße ist eine Combobox, in der Sie für jedes Segment des Backup die Größe in MB festlegen. Sie können eine vordefinierte Größe wählen oder eine eigene Größe zwischen 0 und 2048 angeben. Bei 0 wird nicht segmentiert, das entspricht dem Eintrag Keine.

• **Komprimierung:** 4th Dimension komprimiert die Backups, um Platz zu sparen. Das kann bei großen Datenmengen den Backup-Prozess beträchtlich verlangsamen.
Es gibt folgende Möglichkeiten:
- Keine: Die Dateien werden nicht komprimiert. Das Backup läuft schneller ab, die Archiv-Dateien sind jedoch um einiges größer.
- Schnell (Standard): Das ist ein Mittelmaß zwischen Geschwindigkeit des Backup und Größe des Archivs.
- Kompakt: Hier wird das Archiv maximal komprimiert. Die Dateien beanspruchen den kleinstmöglichen Platz auf der Festplatte, das Backup läuft jedoch um einiges langsamer ab.

• **Interlacing und Redundanz:** 4th Dimension erstellt Archive über spezifische Algorithmen, die auf Operationen zur Optimierung (Interlacing) und Sicherheit (Redundanz) beruhen. Dies können Sie gemäß Ihren Bedürfnissen einstellen. 4th Dimension enthält standardmäßig den Wert **Mittel**. Weitere Möglichkeiten sind **Niedrig**, **Hoch** und **Keine**.

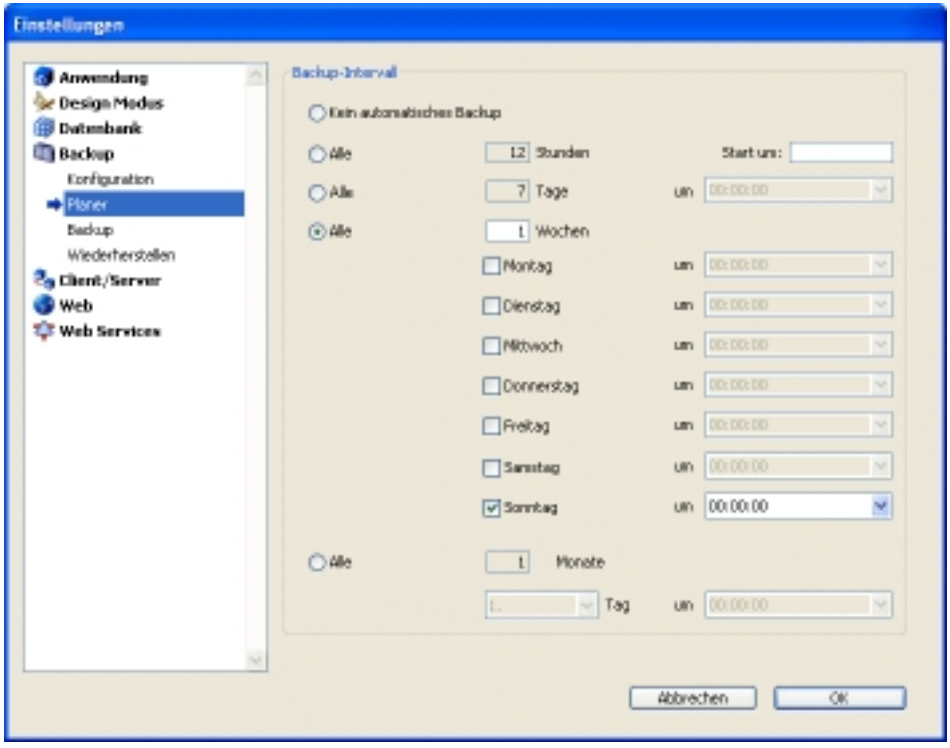
- Interlacing: Damit werden Daten in nicht aufeinanderfolgenden Sektoren gespeichert, um die Lesezeit zu beschleunigen. Diese Speicherphase läuft jedoch langsamer.

- Redundanz: Damit werden die Daten in einer Datei abgesichert, indem dieselbe Information mehrere Male wiederholt wird. Je höher die Redundanzrate, desto besser ist die Sicherheit der Datei. Nachteile sind langsame Speicherung und große Dateien.

Backup nach Zeitplan

Sie können das Backup von Datenbanken, die mit 4th Dimension oder 4D Server geöffnet werden – auch wenn keine Client-Rechner angemeldet sind – automatisch erstellen. Dazu legen Sie einen Zeitplan fest, nach dem Backups erfolgen sollen. Das Zeitintervall lässt sich in Stunden, Tagen, Wochen oder Monaten definieren; 4th Dimension startet dann das Backup automatisch mit den Backup-Einstellungen, die in den Einstellungen der Datenbank auf der Seite Backup festgelegt wurden.

Läuft die Anwendung zum Zeitpunkt des geplanten Backup nicht, erscheint beim nächsten Starten eine Meldung, dass das Backup fehlgeschlagen ist. 4th Dimension wendet dann die entsprechende Konfiguration aus den Einstellungen der Datenbank an. Weitere Informationen dazu finden Sie im Absatz “Wenn Backup fehlschlägt”.
Die Optionen für das Backup nach Zeitplan definieren Sie in den Einstellungen der Datenbank auf der Seite Backup-Intervall:



Sie müssen zuerst eine Zeitspanne festlegen; das können Tage, Wochen, etc. sein. Klicken Sie dazu auf das gewünschte Kontrollkästchen. Dann definieren Sie, wann das Backup ausgeführt wird.

• **Kein automatisches Backup:** Das Backup nach Plan ist inaktiv.

• **Alle X Stunden:** Damit programmieren Sie Backups nach Stunden. Sie können einen Wert zwischen 1 und 24 eingeben.
- Start um: Setzt die Zeit, zu der das Backup zum ersten Mal ausgeführt wird.

• **Alle X Tage um xx:** Damit programmieren Sie Backups nach Tagen. Geben Sie 1 ein, wenn das Backup jeden Tag durchgeführt werden soll. Ist diese Option markiert, müssen Sie auch die Uhrzeit eintragen, zu der das Backup starten soll.

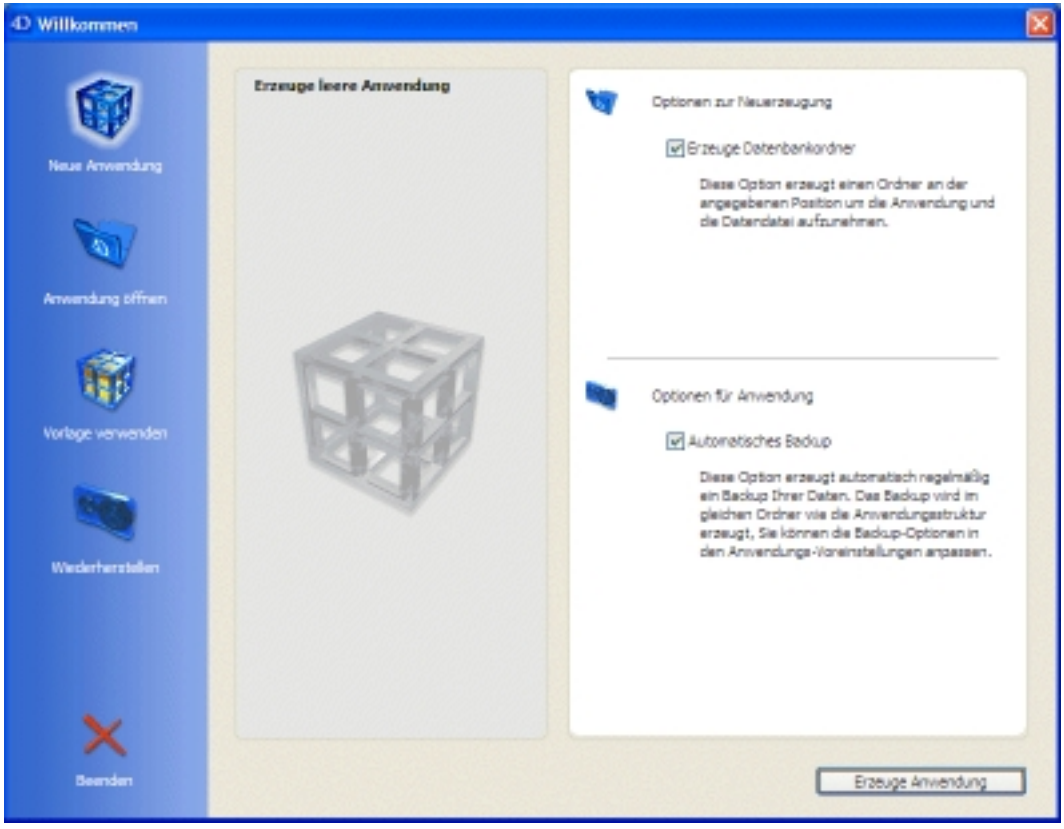
• **Alle X Wochen um xx:** Damit programmieren Sie Backups nach Wochen. Geben Sie 1 ein, wenn das Backup jede Woche durchgeführt werden soll. Ist diese Option markiert, müssen Sie auch den Wochentag und die Uhrzeit eintragen, zu der das Backup starten soll. Sie können bei Bedarf auch mehrere Wochentage ankreuzen.

Sie können z.B. festlegen, dass das Backup zweimal pro Woche läuft, nämlich am Mittwoch und am Freitag.

• **Alle X Monate, X.Tag um xx:** Damit programmieren Sie Backups nach Monaten. Geben Sie 1 ein, wenn das Backup jeden Monat durchgeführt werden soll. Ist diese Option markiert, müssen Sie auch den Wochentag und die Uhrzeit eintragen, zu der das Backup starten soll.

Automatisches Backup

4th Dimension bietet beim Erstellen einer Datenbank eine standardmäßige Backup-Strategie an. Das gewährleistet ein Mindestmaß an Sicherheit für die Datenbank.
Sie können diese Einstellungen verändern, nachdem die Datenbank erstellt ist.
Beim Erstellen einer neuen Datenbank ist standardmäßig die Option automatisches Backup eingestellt. Wollen Sie das nicht, gehen Sie im Willkommensdialog auf die Seite **Neue Anwendung** und deaktivieren rechts unten die Option **Automatisches Backup**:


4D Willkommen
Neue Anwendung
Anwendung öffnen
Vorlage verwenden
Wiederherstellen
Beenden
Erzeuge leere Anwendung
Optionen zur Neuerzeugung
Erzeuge Datenbankordner
Diese Option erzeugt einen Ordner an der angegebenen Position um die Anwendung und die Datendatei aufzunehmen.
Optionen für Anwendung
Automatisches Backup
Diese Option erzeugt automatisch regelmäßig ein Backup Ihrer Daten. Das Backup wird im gleichen Ordner wie die Anwendungsstruktur erzeugt, Sie können die Backup-Optionen in den Anwendungs-Voreinstellungen anpassen.
Erzeuge Anwendung

Die standardmäßige Backup-Strategie beinhaltet folgende Einstellungen:

• Erstellung und Verwendung eines Logbuchs mit Namen DatenbankName.4dl.

• Backup aller Dateien der Datenbank (Daten, Logbuch, Struktur und Benutzerstruktur mit Platzierung neben der Strukturdatei der Datenbank.

• Standardmäßige weitere Optionen (Aufbewahren der letzten 3 Backups, mittlere Raten, etc.).

• Automatisches Backup jeden Sonntag um 00:00:00.

• Alle Optionen zum automatischen Wiederherstellen

Eine Datenbank im Einsatz speichert ständig Anderungen, es werden Datensätze hinzugefügt oder gelöscht. Die Durchführung von Backups in regelmäßigen Abständen ist wichtig, ermöglicht aber nicht, nach einem Zwischenfall die nach dem letzten Backup eingegebenen Daten wiederherzustellen. Dafür bietet 4th Dimension jetzt ein spezifisches Hilfsmittel: das Logbuch. Diese Datei sorgt für permanente Sicherheit der Daten der Datenbank.

Zusätzlich arbeitet 4th Dimension kontinuierlich mit dem Daten-Cache im Speicher. Alle Anderungen an den Daten der Datenbank werden im Cache zwischengespeichert, bevor sie auf die Festplatte geschrieben werden. Das beschleunigt die Operationen der Programme, da der Zugriff auf den Speicher schneller ist als auf die Festplatte. Tritt ein Zwischenfall ein, bevor die im Cache gespeicherten Daten auf die Festplatte geschrieben wurden, müssen Sie beim Wiederherstellen der Datenbank auch das aktuelle Logbuch integrieren.

Einführung

Das von 4D angelegte Logbuch enthält alle Operationen, die in einer Datenbank ausgeführt werden. Somit löst jeder vom Benutzer ausgeführte Schritt zwei Aktionen gleichzeitig aus: Einerseits wird die Anweisung des Benutzers in der Datenbank ausgeführt, zur selben Zeit wird die Aktion im Logbuch mitgeschrieben. Das Logbuch wird unabhängig erstellt, es beeinträchtigt und verlangsamt nicht die Arbeit des Benutzers. Eine Datenbank kann zur selben Zeit immer nur mit einem Logbuch arbeiten.

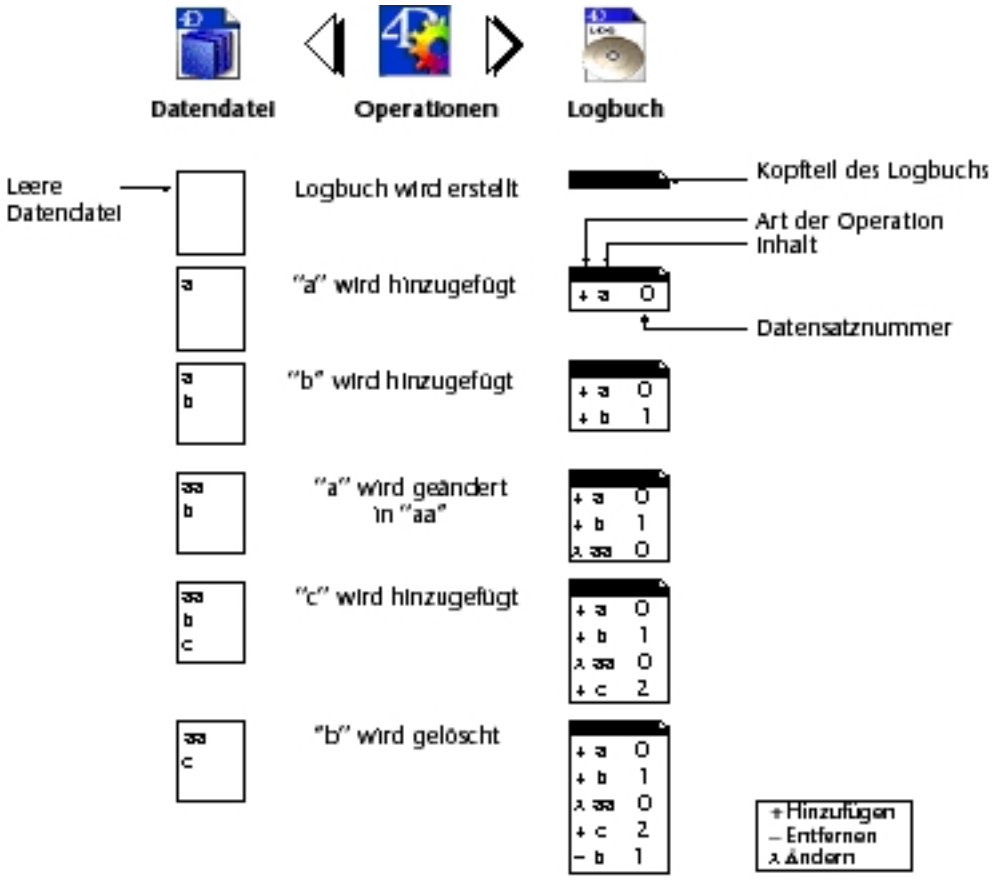
Das Logbuch registriert vier Arten von Operationen:

- Hinzugefügte Datensätze
- Aktualisierte Datensätze
- Gelöschte Datensätze
- Transaktionen

Hinweis zu Transaktionen: Eine Transaktion ist eine Operationsabfolge, die zur selben Zeit ausgeführt wird. Beispiel: Während einer Banküberweisung, die in einer Transaktion ausgeführt wird, müssen Sie sicherstellen, dass Sie in einer Aktion einen Betrag auf einem Konto belasten und auf einem anderen gutschreiben, ohne dass die laufende Operation unterbrochen werden kann. D.h. die Operation wird entweder ganz oder gar nicht ausgeführt, sie kann nie nur teilweise ausgeführt werden. Das erste Konto kann nur belastet werden, wenn gleichzeitig eine Gutschrift auf ein zweites Konto erfolgt.
4th Dimension wertet Transaktionen als eine Operation, sowie Hinzufügen, Aktualisieren oder Löschen.

4th Dimension verwaltet das Logbuch. Es schreibt alle Operationen mit, die die Datendatei betreffen, egal welche Aktion der Benutzer im Einzelplatz von 4th Dimension oder in 4D Client ausführt, in 4D Methoden, 4D Plug-Ins (4D Write, 4D View, 4D Draw, etc.), ein Web Browser, ja sogar 4D Open beim Verwenden von 4D Server.

Nachfolgende Darstellung zeigt, wie das Logbuch arbeitet:



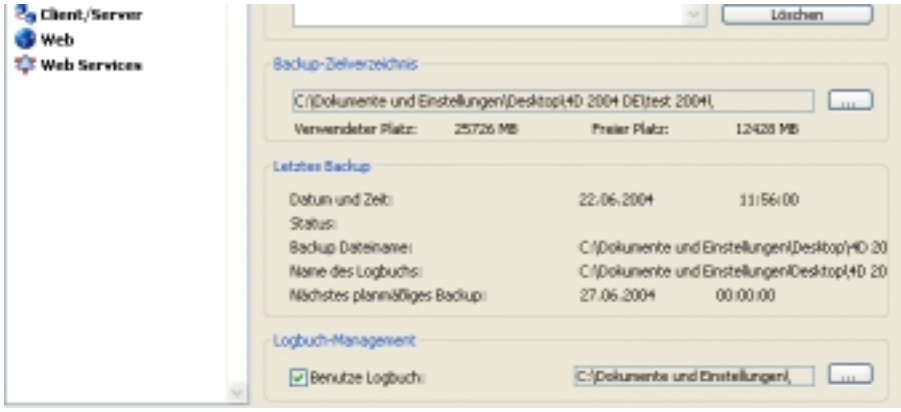
Das aktuelle Logbuch wird automatisch mit der aktuellen Datendatei gesichert. Das hat folgende Vorteile:

• Wichtig ist, dass das Volume, in welchem das Logbuch gespeichert wird, nicht voll wird. Ohne Backup würde das Logbuch mit der Zeit beträchtlich anwachsen und u.U. den gesamten Platz im Verzeichnis belegen. 4th Dimension bzw. 4D Server schliesst für jeden Backup-Satz das aktuelle Logbuch und startet sofort eine neue leere Datei, so dass auf dem Volume genügend Platz frei bleibt. Das alte Logbuch wird je nach Backup-Einstellungen archiviert oder entfernt.

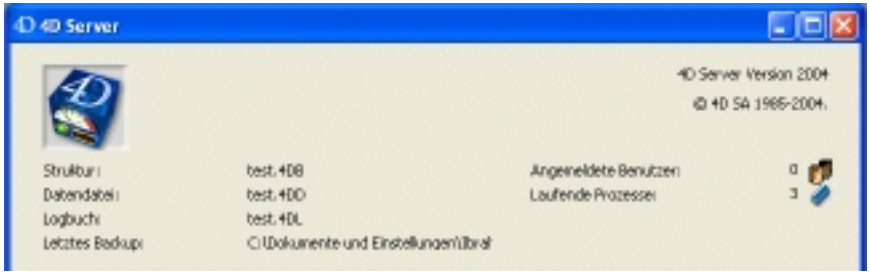
• Behalten Sie die Logbücher zu den entsprechenden Backups, damit Sie eine Datenbank später analysieren oder reparieren können. Das Logbuch kann nur in die Datenbank integriert werden, zu der es gehört. Sie müssen Backups und Logbücher zusammen archivieren, damit Sie das Logbuch bei Bedarf korrekt in das Backup einbinden können.

Logbuch erstellen

Jede mit 4th Dimension 2004 erstellte Datenbank führt automatisch ein Logbuch: Es wird automatisch angelegt, wenn im Willkommensdialog auf der Seite **Erzeuge** die Option **Automatisches Backup** markiert ist.
Das Logbuch hat den Namen DatendateiName.4DL und wird in den Ordner gelegt, der die Strukturdatei der Datenbank enthält. Sie können jederzeit prüfen, ob Ihre Datenbank ein Logbuch führt. Dazu gehen Sie in den Einstellungen der Datenbank auf die Seite **Konfiguration** und sehen nach, ob die Option **Benutze Logbuch** markiert ist.



Mit 4D Server erscheint der Name des Logbuchs auch im Fenster des Server-Rechners:



Ist diese Option inaktiv oder verwenden Sie eine Datenbank ohne Logbuch und wollen dann eine Backup-Strategie mit Logbuch einrichten, müssen Sie ein Logbuch anlegen.

1. Um ein Logbuch anzulegen, gehen Sie in den Einstellungen der Datenbank unter dem Thema **Backup** auf die Seite **Konfiguration** und markieren Sie am unteren Rand die Option **Benutze Logbuch.**

Hinweis zur Kompatibilität: Diese Option ersetzt die Option “Logbuch erforderlich” aus früheren Versionen von 4th Dimension.

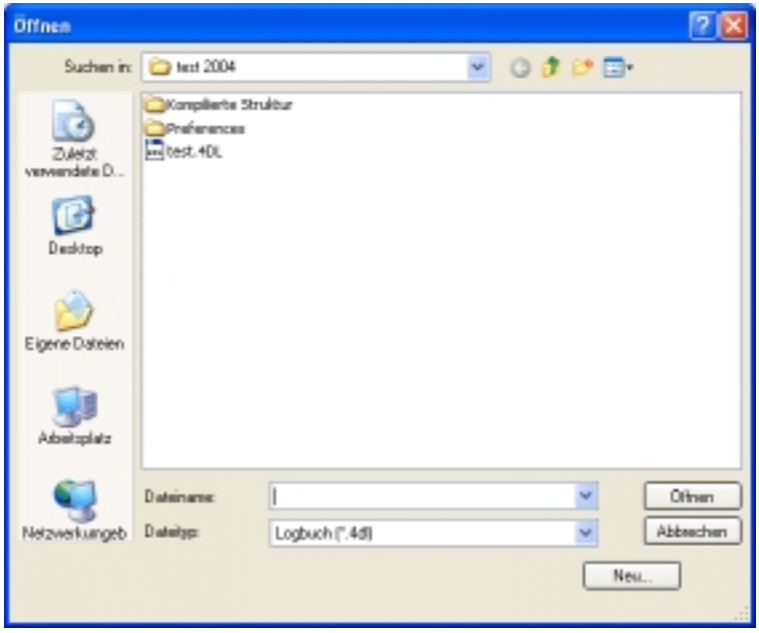
Damit Sie ein Logbuch erstellen können, muss die Datenbank sich in einer der folgenden Situationen befinden:

• Die Datendatei ist leer,

• Sie haben gerade ein Backup der Datenbank ausgeführt und es hat noch keine Anderungen an den Daten gegeben.

In allen anderen Fällen zeigt das Programm eine Meldung, ob Sie ein Backup durchführen wollen. Diese Vorsichtsmaßnahme ist notwendig, denn, damit Sie die Datenbank nach einem Zwischenfall wiederherstellen können, benötigen Sie eine Kopie der Datenbank, in die Sie die im Logbuch gespeicherten Operationen integrieren können.

Trifft eine der o.a. Bedingungen zu, zeigt das Programm einen Dialog zum Öffnen einer Datei oder zum Anlegen eines neuen Logbuchs:



2. Klicken Sie auf die Schaltfläche **Neu.**

Auf dem Bildschirm erscheint ein Standard Sichern-Dialog. Das Logbuch heißt standard-mäßig DatendateiName.4DL.

3. Behalten Sie den Standardnamen oder benennen Sie ihn um und wählen dann die Platzierung der Datei.

Wir empfehlen, das Logbuch nicht auf dieselbe Platte wie die Datenbank zu legen. Geht diese Platte kaputt, können Sie in jedem Fall das Logbuch aufrufen.

4. Klicken Sie auf die Schaltfläche **Öffnen**.

Zugriffspfad und Name des aktuellen Logbuchs erscheinen nun in den Einstellungen der Datenbank im Bereich “Benutze Logbuch”.
Ab jetzt werden alle Operationen in den Daten im Logbuch mitgeschrieben. Dieses wird in Zukunft immer zusammen mit der Datenbank verwendet.

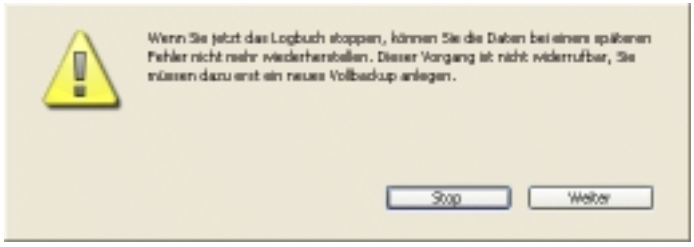
• Erstellen Sie eine neue Datendatei, müssen Sie auch ein neues Logbuch anlegen.

• Sie müssen ein anderes Logbuch einrichten, wenn Sie eine andere Datendatei öffnen, die nicht mit einem Logbuch verknüpft ist oder wenn das Logbuch fehlt. Dazu klicken Sie auf die Schaltfläche [...] neben dem Eingabebereich mit dem Zugriffspfad zum Logbuch.

Logbuch stoppen

Wollen Sie das Mitschreiben der Operationen für das aktuelle Logbuch stoppen, gehen Sie in die Einstellungen der Datenbank zum Thema **Backup** und deaktivieren Sie auf der Seite **Konfiguration** die Option "Benutze Logbuch".

Auf dem Bildschirm erscheint folgende Meldung:

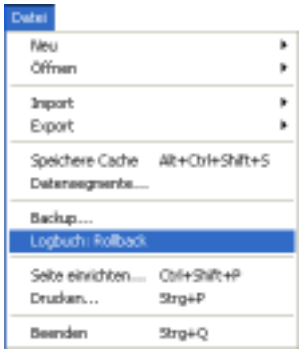

Wenn Sie jetzt das Logbuch stoppen, können Sie die Daten bei einem späteren Fehler nicht mehr wiederherstellen. Dieser Vorgang ist nicht widerrufbar, Sie müssen dazu erst ein neues Vollbackup anlegen.
Stop
Weiter

Klicken Sie auf die Schaltfläche **Stop**, wird das aktuelle Logbuch sofort geschlossen, d.h. das Dialogfenster Einstellungen muss danach nicht bestätigt werden.
Wollen Sie das aktuelle Logbuch schliessen, weil es zu groß ist, sollten Sie von der Datendatei ein Backup ausführen, welches dann auch das Logbuch sichert.

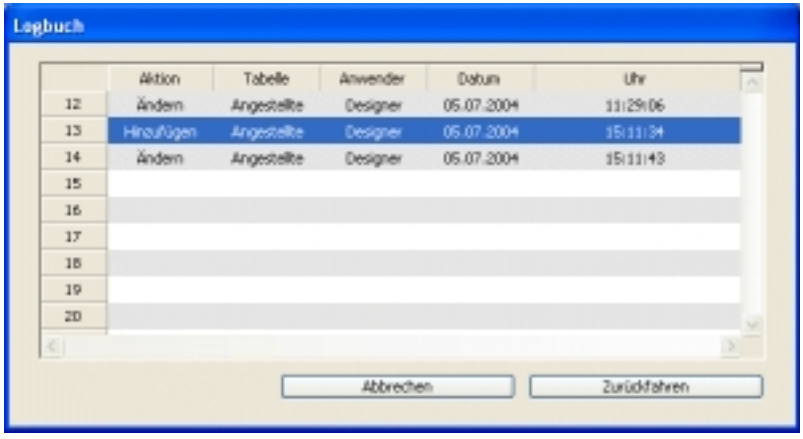
Logbuch analysieren

4th Dimension bietet eine Funktion, über die Sie den Inhalt des aktuellen Logbuchs ansehen können. Das ist hilfreich, um die Verwendung der Datenbank zu analysieren oder um die Operationen herauszufinden, die Fehler oder Fehlfunktionen verursachen. Läuft die Datenbank im Client/Server-Betrieb, können Sie die Operationen jedes Client-Rechners überprüfen.
Auf diese Weise können Sie die Operationen in den Daten der Datenbank rückgängig machen.

Um das aktuelle Logbuch anzusehen, wählen Sie in 4th Dimension oder 4D Server im Menü **Datei/Ablage** den Befehl **Logbuch:Rollback**. Diese Funktion ist auf dem Client nicht verfügbar:



Auf dem Bildschirm erscheint folgendes Fenster:



Jede im Logbuch mitgeschriebene Operation erscheint in einer Zeile. Die Spalten enthalten verschiedene Informationen zur Operation. Sie können die Standardanzeige in den Spalten verändern, z.B. über das Kontextmenü des Fensters die Werte der Felder anzeigen.
Über diese Angaben können Sie die Quelle und den Kontext jeder Operation herausfinden.

Aktion: Art der ausgeführten Operation in den Daten. Diese Spalte kann folgende Einträge enthalten: Hinzufügen, Löschen, Andern, Transaktion. Weitere Informationen dazu finden Sie im Absatz “Einleitung” auf Seite 229.

Tabelle: Tabelle, zu der die bearbeiteten Datensätze gehören.

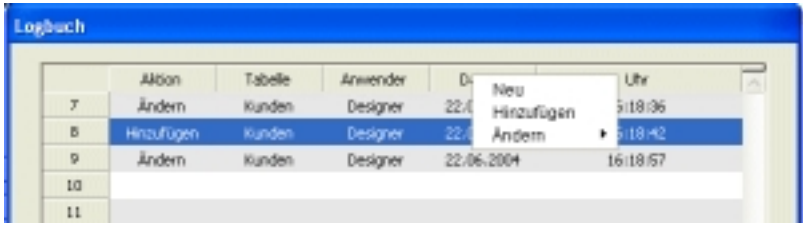
Anwender: Name des Benutzers, der die Operation ausgeführt hat; im Client/Server-Betrieb erscheint der Name des Client-Rechners, im Einzelplatz die Benutzerkennung. Sind keine 4th Dimension Kennwörter aktiviert, zeigt diese Spalte
Designer an.
Wurde die Operation mit 4D Open ausgeführt, erscheint die Bezeichnung “4D Open”.

Datum und **Zeit:** Datum und Uhrzeit der Ausführung.

Spaltenanzeige setzen

Über das Kontextmenü des Fensters Logbuch können Sie die Anzeige in den Spalten an Ihre Bedürfnisse anpassen. Sie können Spalten hinzufügen oder entfernen, den Inhalt einer Spalte für das aktuelle Fenster ersetzen. Der angepasste Fensterinhalt wird nach Schließen des Fensters wieder zurückgesetzt.

Das Kontextmenü erscheint, wenn Sie auf einen Spaltentitel klicken:



Es gibt folgende Einträge:

Neu: Damit fügen Sie rechts neben den vorhandenen Spalten eine Spalte hinzu. Die Spalte ist standardmäßig leer und hat den Namen “Feld”. Sie kann einem Feld der Datenbank zugeordnet werden (siehe nächster Abschnitt).

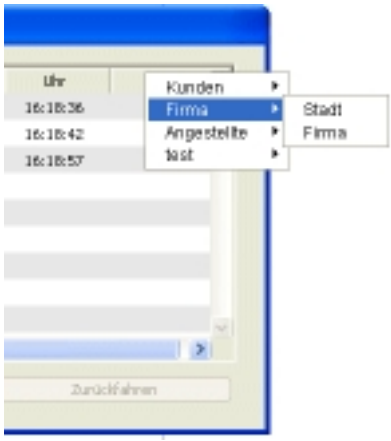
Entfernen: Damit entfernen Sie eine Spalte aus dem Fenster.

Andern: Damit ersetzen Sie den Inhalt der aktuellen Spalte. Sie können entweder eine Standardinformation wie Aktion, Tabelle, etc. wählen oder Feld, um eine Spalte hinzuzufügen, die die Werte eines Feldes der Datenbank anzeigt.

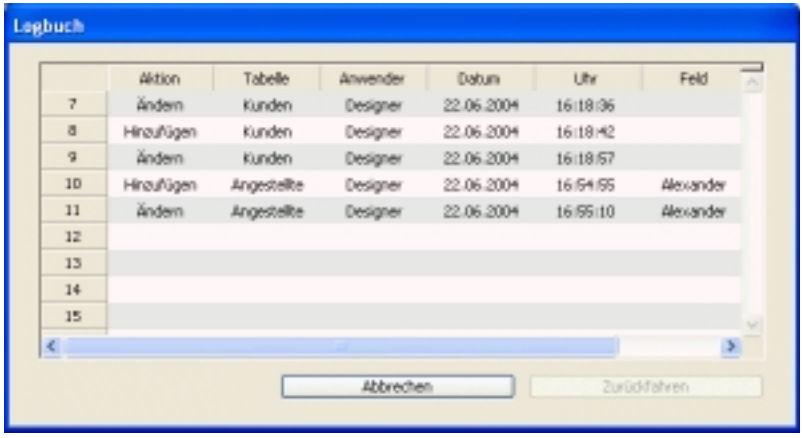
Feldwerte ansehen

Sie können einer Spalte im Fenster Logbuch ein Feld der Datenbank zuweisen, so dass für jede mitgeschriebene Operation der aktuelle Wert des Feldes angezeigt wird. Das ist besonders hilfreich, wenn Sie z.B. herausfinden wollen, wann ein bestimmter Datensatz hinzugefügt wurde.
Um eine leere Spalte vom Typ Feld hinzuzufügen, wählen Sie im Kontextmenü des Fensters den Eintrag **Hinzufügen** oder **Ersetzen** (siehe oben).

Klicken Sie auf den Spaltentitel, erscheint ein hierarchisches Menü mit den Tabellen und Feldern der Datenbank:

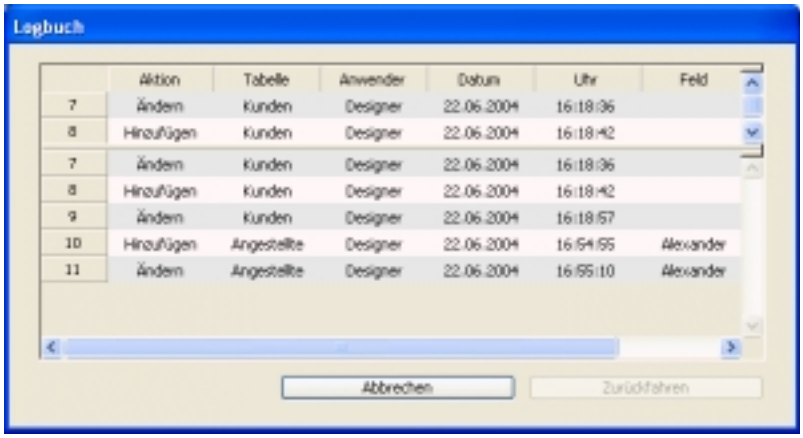


Wählen Sie das entsprechende Feld. Die Werte erscheinen dann für jeden hinzugefügten oder geänderten Datensatz:



Ansichten

Sie können im Fenster des Logbuchs Ansichten erstellen. Dazu klicken Sie auf das Splitter-Symbol rechts oben und ziehen es nach unten. Das Fenster sieht jetzt folgendermaßen aus:



Um mehrere Ansichten zu erstellen, wiederholen Sie diese Operation so oft wie erforderlich. Sie können auch eine Ansicht vor eine bereits erstellte setzen, wenn Sie deren Splitter ziehen.
Um eine Ansicht wieder aufzuheben, ziehen Sie den Splitter am unteren Rand nach oben oder doppelklicken Sie auf den Splitter.

Referenz

Backup verwalten

Mit dem in 4th Dimension integrierten Backup-Modul können Sie komplette Datenteile der Datenbank bei Zwischenfällen, egal welcher Art, wiederherstellen.

Zwischenfälle und Diagnose

Es gibt zwei Hauptkategorien für Zwischenfälle:

• **Unvorhergesehener Abbruch der Datenbank in Betrieb**
Dieser Zwischenfall kann bei Stromausfall, Fehler im Betriebssystem, o.ä. passieren. Je nachdem, welchen Status der Daten-Cache im Augenblick des Zwischenfalls hat, sind zum Wiederherstellen der Datenbank unterschiedliche Operationen erforderlich:

- War der Cache leer, öffnet sich die Datenbank ganz normal. Alle Anderungen in der Datenbank wurden gespeichert. In diesem Fall ist keine besondere Operation erforderlich.

- Enthält der Cache Operationen, ist die Datenbank intakt. Sie müssen jedoch das aktuelle Logbuch integrieren, um den Zustand im Moment des Abbruchs wiederherzustellen.

- Sollte der Cache gerade auf die Festplatte geschrieben werden, ist die Datendatei wahrscheinlich beschädigt. In diesem Fall müssen Sie das letzte Backup wiederherstellen und das aktuelle Logbuch integrieren.

• **Verlust von Dateien der Datenbank**
Dieser Zwischenfall kann bei fehlerhaften Sektoren auf der Festplatte, bei einem Virus, bei Bedienungsfehlern, o.ä. passieren.
Hier müssen Sie das letzte Backup wiederherstellen und das aktuelle Logbuch integrieren.
Um herauszufinden, ob eine Datenbank bei einem Zwischenfall beschädigt wurde, starten Sie einfach die Datenbank wieder mit 4th Dimension. Das Programm führt selbst eine Prüfung durch und listet auf, welche Operationen zum Wiederherstellen ausgeführt werden müssen.
Im automatischen Modus werden diese Operationen direkt ausgeführt, ohne Eingreifen durch den Benutzer. Weitere Informationen dazu finden Sie im nächsten Absatz.
Haben Sie eine reguläre Backup-Strategie eingerichtet, finden Sie über die 4th Dimension Tools zum Wiederherstellen in der Regel die Datenbank in genau dem Zustand vor dem Zwischenfall vor. Zwei Haupt-Tools stehen zur Verfügung:

• Wiederherstellen des Backup
• Integration des Logbuchs

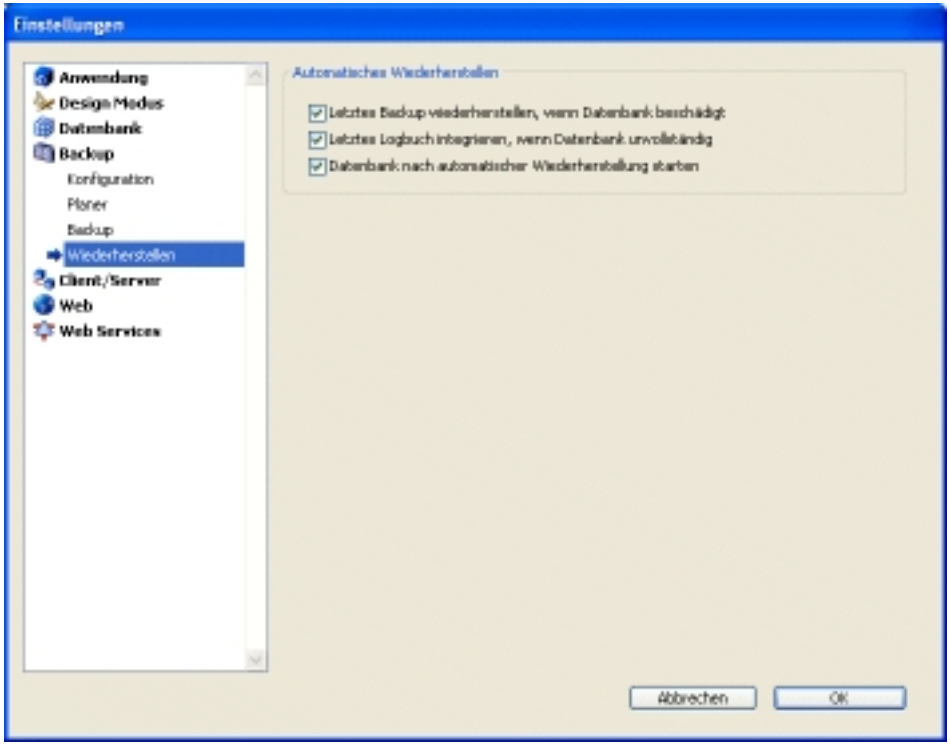
Beide Funktionen lassen sich auch miteinander kombinieren und automatisieren.

Automatisches Wiederherstellen

4th Dimension startet nach einem Zwischenfall automatisch die Prozedur zum Wiederherstellen der Datenbank. Das sind folgende Abläufe:

- Automatisches Wiederherstellen
- Automatische Integration des Logbuchs
- Automatisches Starten der Datenbank nach Wiederherstellen

Sie können diese Abläufe in den Einstellungen der Datenbank auf der Seite **Wiederherstellen** unter dem Thema **Backup** einzeln aktivieren/deaktivieren:



Letztes Backup wiederherstellen, wenn Datenbank beschädigt

Mit dieser Option startet das Programm automatisch die Wiederherstellung der Daten des letzten gültigen Backup der Datenbank, wenn beim Hochfahren der Datenbank eine Unregelmäßigkeit festgestellt wird. Der Benutzer selbst muss nichts ausführen; die Operation wird im Logbuch des Backup vermerkt. Weitere Informationen dazu finden Sie im Abschnitt Backup Journal.

Hinweise:
• Beim Wiederherstellen werden Dateien aus einer bzw. mehreren Backup-Dateien entnommen und erneut für die Anwendung, die sie erstellt hat, lesbar gemacht.
• Beim automatischen Wiederherstellen wird nur die Datendatei wiederherstellt. Soll das auch für die angefügten Dateien oder die Strukturdatei gelten, müssen Sie eine manuelle Wiederherstellung ausführen.

Bei automatischer Wiederherstellung werden folgende Schritte ausgeführt:

• 4th Dimension gibt zuerst der beschädigten Datendatei einen anderen Namen.

• 4th Dimension entnimmt die Datendatei des letzten Backup und speichert sie an der Stelle der vorigen Datei.

• Ist die Option "Letztes Logbuch integrieren, wenn Datenbank unvollständig" markiert, integriert 4th Dimension bei Bedarf das Logbuch. Weitere Informationen dazu finden Sie im Absatz "Letztes Logbuch integrieren, wenn Datenbank unvollständig".

• Ist die Option "Datenbank nach automatischer Wiederherstellung starten" markiert, startet 4th Dimension die wiederhergestellte Datenbank. Weitere Informationen dazu finden Sie im Absatz "Datenbank nach automatischer Wiederherstellung starten".

Letztes Logbuch integrieren, wenn Datenbank unvollständig

Ist diese Option markiert, integriert das Programm automatisch das Logbuch, wenn eine wiederhergestellte Datenbank geöffnet wird. Der Benutzer selbst muss nichts ausführen.

• Beim Öffnen einer Datenbank wird das aktuelle Logbuch automatisch integriert, wenn 4th Dimension feststellt, dass die im Logbuch gespeicherten Operationen in den Daten nicht vorhanden sind. Das passiert z.B. bei Stromausfall, wenn es Operationen im Daten-Cache gibt, die noch nicht in den Hauptspeicher übertragen wurden.

• Hat beim Wiederherstellen einer Datenbank das aktuelle Logbuch bzw. Logbuch des Backup dieselbe Nummer wie das Backup, das im gleichen Ordner gespeichert ist, prüft 4th Dimension seinen Inhalt. Enthält es Operationen, die nicht in der Datendatei enthalten sind, wird es integriert.

Auf Benutzerseite erscheint kein Dialogfenster, die Operation läuft vollkommen automatisch ab. Ziel dabei ist, die Verwendung so einfach wie möglich zu machen. Die Operation wird im Logbuch des Backup vermerkt. Weitere Informationen dazu finden Sie im Abschnitt Backup Journal.

Datenbank nach automatischer Wiederherstellung starten

Ist diese Option markiert, öffnet 4th Dimension automatisch eine Datenbank, die nach Abschließen der Prozedur wiederhergestellt wurde.
Bei dieser Option müssen Sie darauf achten, dass alle Elemente, die zum Betreiben der Datenbank notwendig sind, nach der Wiederherstellung verfügbar sind. Benötigt eine Datenbank z.B. auf derselben Ebene wie die Strukturdatei spezifische Dateien, müssen diese in die Backups integriert werden, so dass sie gleichzeitig mit der Datenbank wiederher gestellt werden können. Arbeitet die Datenbank mit Plug-Ins, ist es besser, diese auf derselben Ebene wie das Programm 4th Dimension zu installieren.

Diese drei Optionen zum Wiederherstellen im Zusammenspiel mit den Optionen für Backup ermöglichen ein halb- oder vollautomatisches System zum Rückführen auf den Stand der Datenbank vor dem Zwischenfall.

Halbautomatisch: Der Administrator muss nur die Datenbank erneut starten, 4th Dimension führt automatisch die notwendigen Operationen zur Wiederherstellung aus und startet eine zuverlässige Version der Anwendung (aktualisiert).

Automatisch (nur 4D Server): Konfigurieren Sie die Datenbank als Service, der bei Systemstart aktiviert wird, wird er beim Hochfahren des Rechners automatisch erneut gestartet. Weitere Informationen dazu finden Sie im Absatz “4D Server als MacOS X Service registrieren” auf Seite 24 bzw. für Windows im Handbuch 4D Server.?? 4th Dimension führt automatisch die notwendigen Operationen zur Wiederherstellung aus.

Diese Art System ist besonders gut an 4D Server Datenbanken angepasst. Beachten Sie, dass jede ausgeführte Operation automatisch im Logbuch des Backup mitgeschrieben wird. Weitere Informationen dazu finden Sie im Absatz “Letztes Logbuch integrieren, wenn Datenbank unvollständig”.

Backup manuell wiederherstellen

Sie können den Inhalt eines Archivs, das vom Backup-Modul erstellt wurde, manuell wiederherstellen. Manuelles Wiederherstellen ist z.B. erforderlich, um den Archivinhalt im Ganzen zu reproduzieren (Strukturdateien und/oder angehängte Dateien) oder zur Ausführung von Suchläufen in den Archiven. Wiederherstellen kann auch mit Integration des aktuellen Logbuchs erfolgen.

Hinweis: Sie können 4th Dimension so einstellen, dass die Datenbank bei einen Zwischenfall, egal welcher Art, wiederhergestellt wird. Weitere Informationen dazu finden Sie im Abschnitt Daten wiederherstellen.

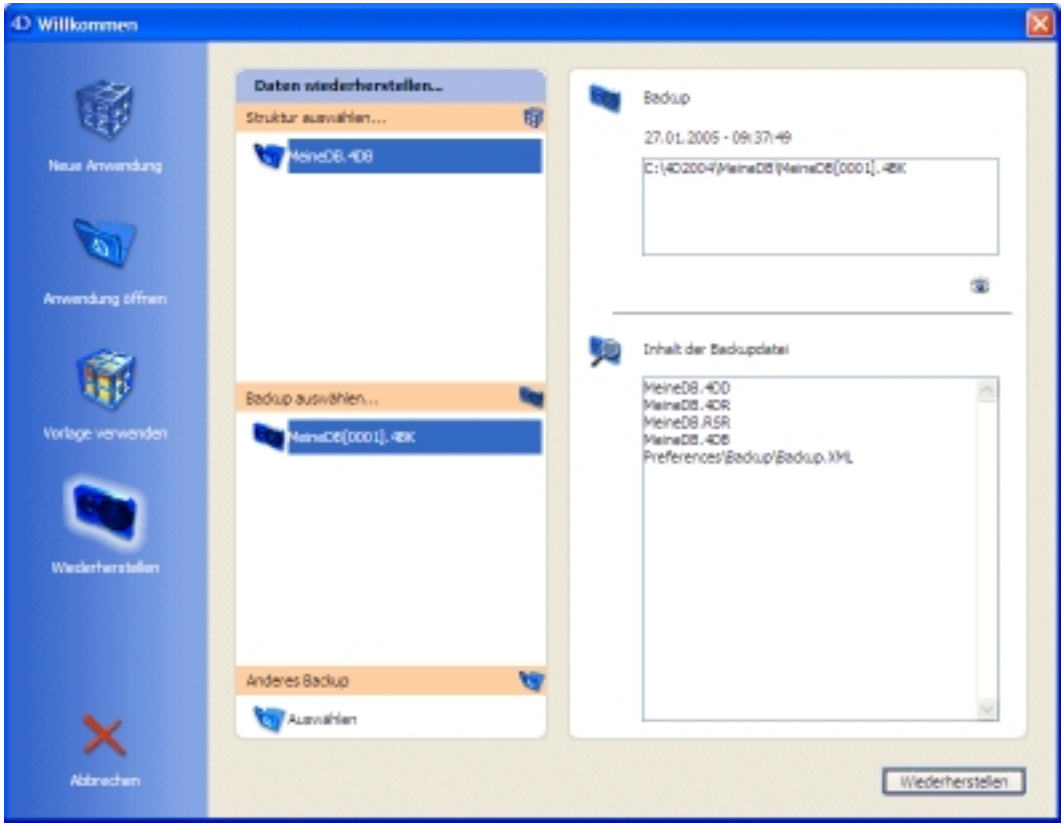
Die 4D Anwendung selbst (4th Dimension, 4D Server, Anwendung mit einkompilierter 4D Runtime Volume License, etc.) führt die Wiederherstellung des Backup über die Seite **Wiederherstellen** im Willkommensdialog aus.

Um eine Datenbank manuell wiederherzustellen:

1. Klicken Sie im Willkommensdialog am rechten Rand auf die Schaltfläche **Wiederherstellen** (nur 4th Dimension und 4D Server)
ODER
Führen Sie in einer 4th Dimension Methode den Befehl RESTORE aus (alle 4D Programme)
ODER
Ziehen Sie die Backup-Datei für die Wiederherstellung per Drag&Drop auf das Icon der Anwendung (alle 4D Programme).

Im letzten Fall erscheint ein Standard Sichern-Dialog, in dem Sie angeben, wo die Dateien wiederhergestellt werden. Gehen Sie direkt zu Schritt 3.

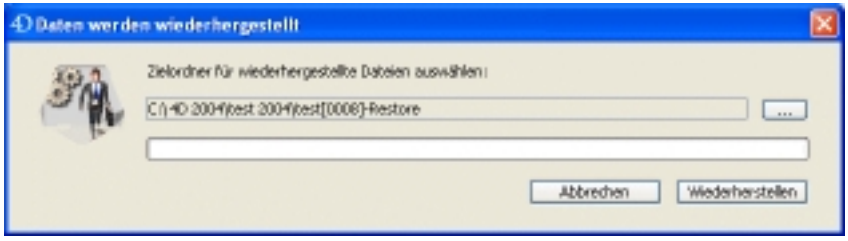
2. Im Bereich “Struktur auswählen” wählen Sie die Datenbank, dann das Backup für die Wiederherstellung und klicken auf die Schaltfläche **Wiederherstellen**:



Der rechte Teil des Dialogfensters zeigt den Zugriffspfad und den Inhalt des gewählten Archivs. Bei segmentiertem Backup erscheint der Zugriffspfad des ersten Segments.

Um ein anderes Archiv wiederherzustellen, klicken Sie auf **Anderes Backup**.

Ein Dialogfenster erscheint, in dem Sie angeben, wo die Dateien wiederherstellt werden sollen.



4th Dimension stellt die Dateien standardmäßig in einem Ordner mit Namen „Archivname-Restore“ wieder her, der neben dem Archiv liegt. Sie können auf die Schaltfläche [...] klicken, um eine andere Platzierung festzulegen.

3. Andern Sie bei Bedarf die Platzierung der wiederhergestellten Dateien und klicken Sie auf die Schaltfläche **Wiederherstellen**.

4th Dimension setzt alle Backup-Dateien an den festgelegten Ort.

Hat das aktuelle Logbuch bzw. Logbuch des Backup dieselbe Nummer wie das Backup, das im gleichen Ordner gespeichert ist, prüft 4th Dimension seinen Inhalt. Enthält es Operationen, die nicht in der Datendatei enthalten sind, fragt das Programm, ob Sie diese Operationen integrieren wollen.

Hinweis: Die Integration erfolgt automatisch, wenn die Option zum Integrieren des Logbuchs markiert ist. Weitere Informationen dazu finden Sie im Absatz “Letztes Logbuch integrieren, wenn Datenbank unvollständig”.

4. (Optional) Klicken Sie auf die Schaltfläche **OK**, um das Logbuch in die wiederhergestellte Datenbank zu integrieren.

Sind die Operationen zum Wiederherstellen und Integrieren korrekt abgelaufen, zeigt 4th Dimension die Meldung, ob Sie die Datenbank öffnen möchten.

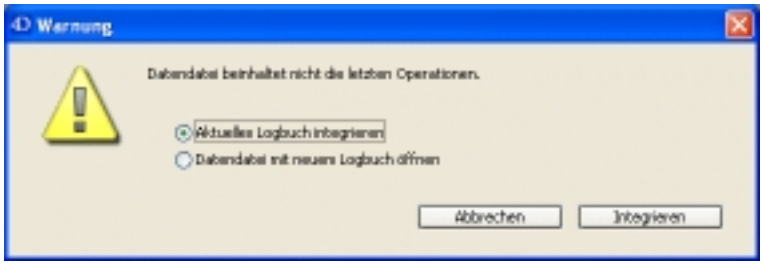
5. Klicken Sie auf die Schaltfläche **OK**, um die wiederhergestellte Datenbank zu öffnen
ODER
auf die Schaltfläche **Abbrechen**, um das Programm 4th Dimension zu beenden.

Hinweis: Die Datenbank startet automatisch, wenn die Option zum automatischen Starten markiert ist. Weitere Informationen dazu finden Sie im Absatz “Datenbank nach automatischer Wiederherstellung starten”.

4th Dimension legt während der Wiederherstellung alle Backup-Dateien in den Zielordner, egal an welcher Stelle auf der Festplatte die Orginaldateien beim Starten des Backup liegen. So finden Sie die Dateien leichter.

Logbuch manuell integrieren

Ist die Option für automatisches Integrieren des Logbuchs nicht aktiv, erscheint beim Öffnen der Datenbank eine Meldung, wenn 4th Dimension feststellt, dass das Logbuch mehr Operationen enthält als in der Datenbank ausgeführt wurden.



Damit dieser Vorgang funktioniert, muss 4th Dimension auf das Logbuch an dessen aktueller Platzierung zugreifen können.
Sie entscheiden selbst, ob Sie das aktuelle Logbuch integrieren wollen.

Operationen rückgängig machen

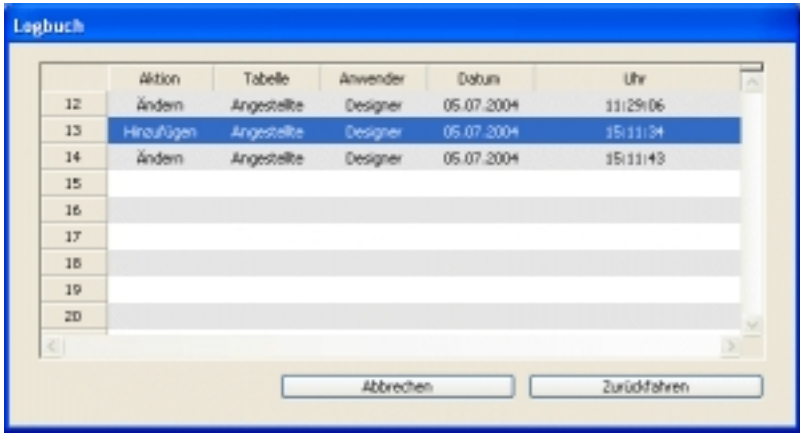
4th Dimension bietet eine Funktion, über die sich Operationen, die im Logbuch aufgeführt sind, rückgängig machen lassen. Der Vorgang ähnelt dem Widerrufen in mehreren Stufen. Er ist besonders hilfreich, wenn versehentlich ein Datensatz aus der Datenbank gelöscht wurde.
Das ist nur durchführbar, wenn die Datenbank ein Logbuch hat.

Um eine Datenbank in einem früheren Stadium wiederherzustellen:

1. Wählen Sie in 4th Dimension oder 4D Server im Menü **Datei/Ablage** den Befehl **Prüfe Logbuch** (ist auf Client-Seite nicht verfügbar).
Es erscheint ein Dialogfenster mit den Operationen im Logbuch. Weitere Informationen dazu finden Sie im Abschnitt Logbuch verwalten.

2. Wählen Sie die Zeile, nach der alle Operationen annulliert werden sollen.
Die Operation der gewählten Zeile bleibt als letzte erhalten. Wollen Sie z.B. einen Löschvorgang aufheben, wählen Sie die direkt davor liegende Operation.

Diese bleibt dann erhalten, alle nachfolgenden Operationen werden zurückgefahren.



3. Klicken Sie auf die Schaltfläche **Zurückfahren**.
Auf dem Bildschirm erscheint ein Dialogfenster zum Bestätigen.

4. Klicken Sie auf die Schaltfläche **OK**.
Die Daten werden dann in dem Stadium wiederhergestellt, das zum Zeitpunkt der gewählten Operation galt.

Arbeitsweise

Klickt der Benutzer auf die Schaltfläche **Zurückfahren**, schliesst 4th Dimension die aktuelle Datenbank und stellt das letzte Backup der Daten der Datenbank wieder her. Die wiederher gestellte Datenbank wird dann geöffnet. 4th Dimension integriert die Operationen des Logbuchs bis zur gewählten Operation.
Wurde die Datenbank noch nicht gesichert, startet 4th Dimension mit einer leeren Daten-datei.

Damit das Nachvollziehen und Überprüfen von Backups einfacher ist, schreibt das Backup-Modul einen Abriss jeder ausgeführten Operation in eine spezielle Datei, die eine Art Aktivitätsjournal ist. Hier werden – wie in einem Boardbuch – alle Operationen der Datenbank (Backup, Wiederherstellen, Integration des Logbuchs) eingetragen, und ob sie nach Zeitplan oder manuell ausgeführt wurden. Zusätzlich erscheinen Datum und Uhrzeit der Ausführung.

Das Journal macht je nach Bereich des Backup folgende Angaben:

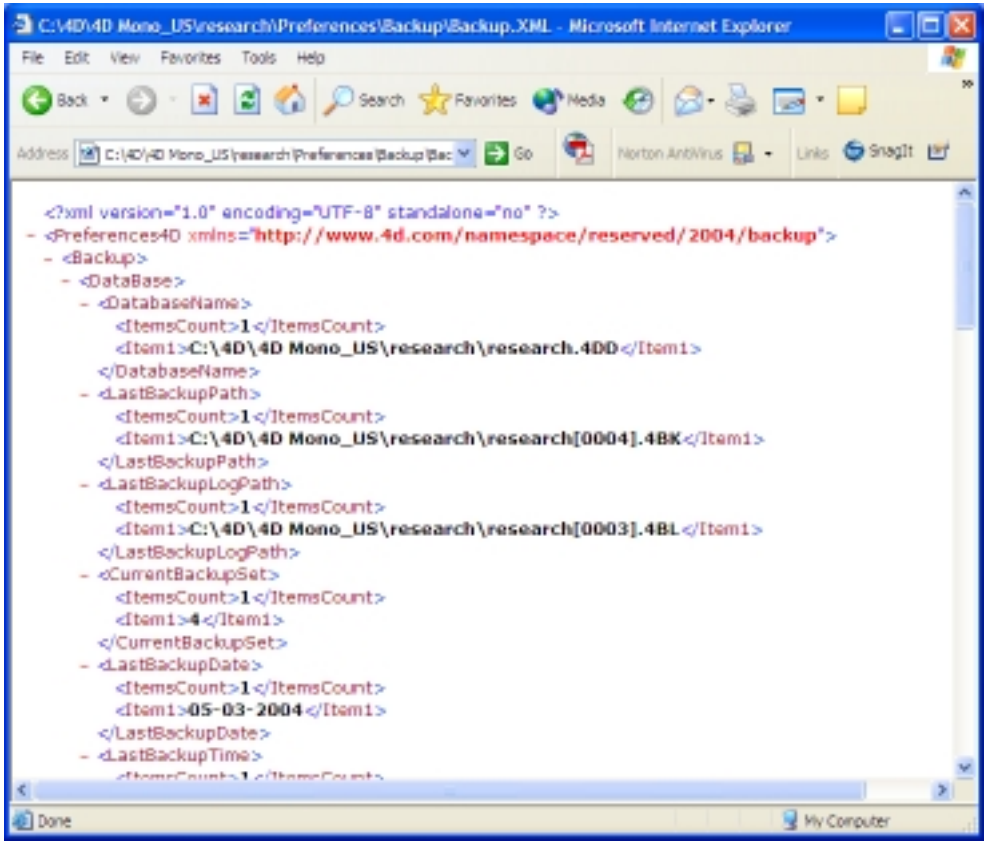
Für jedes Archiv:
- Zugriffspfad (für das Archiv, das erstellte oder gelöschte Logbuch)
- Größe
- Archivinhalt (Datendatei, Strukturdatei, Strukturdatei des Benutzers, Index-Dateien, Logbuch)

Für jeden Eintrag im Archiv:
- Dateityp
- Originalgröße der Datei
- Datum/Uhrzeit (Startzeit, planmäßige Zeit, Endzeit)

Das Backup Journal hat den Namen “Journal.txt” und liegt auf derselben Ebene wie die Datendatei der Datenbank. Es lässt sich mit einem beliebigen Texteditor öffnen. Zur besseren Übersicht enthält das Journal Einrückungen, jede Informationszeile endet mit einer Zeilenschaltung.

Alle Einstellungen für Backup und Wiederherstellen in 4th Dimension sind als eigenständige XML-Datei verfügbar. 4th Dimension verwendet die darin enthaltenen Daten zur Anzeige der Optionen in den Einstellungen der Datenbank und in dem Moment, wenn ein Backup startet.
Sie können die Einstellungen für Backup über XML Befehle in 4th Dimension oder einen beliebigen XML Editor aufrufen und verändern. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* im Kapitel "XML".

Das ermöglicht Entwicklern, Backup-Einstellungen per Programmierung zu verwalten, insbesondere bei Anwendungen mit einkompilierter 4D Runtime Volume License:



Diese Datei enthält zwei Einstellungsblöcke:

• **Allgemeine Einstellungen:** Allgemeine und erweiterte Einstellungen des Backup.

• **Spezifische Einstellungen der Datendatei:** Einstellungen des durchgeführten Backup (Datum und Uhrzeit des letzten Backup, Datum und Uhrzeit des nächsten Backup, Platzierung und Nummer der letzten Backup-Datei).

Die XML-Datei mit den Einstellungen für Backup und Wiederherstellen hat den Namen Backup.XML. Sie wird automatisch an folgender Stelle angelegt:

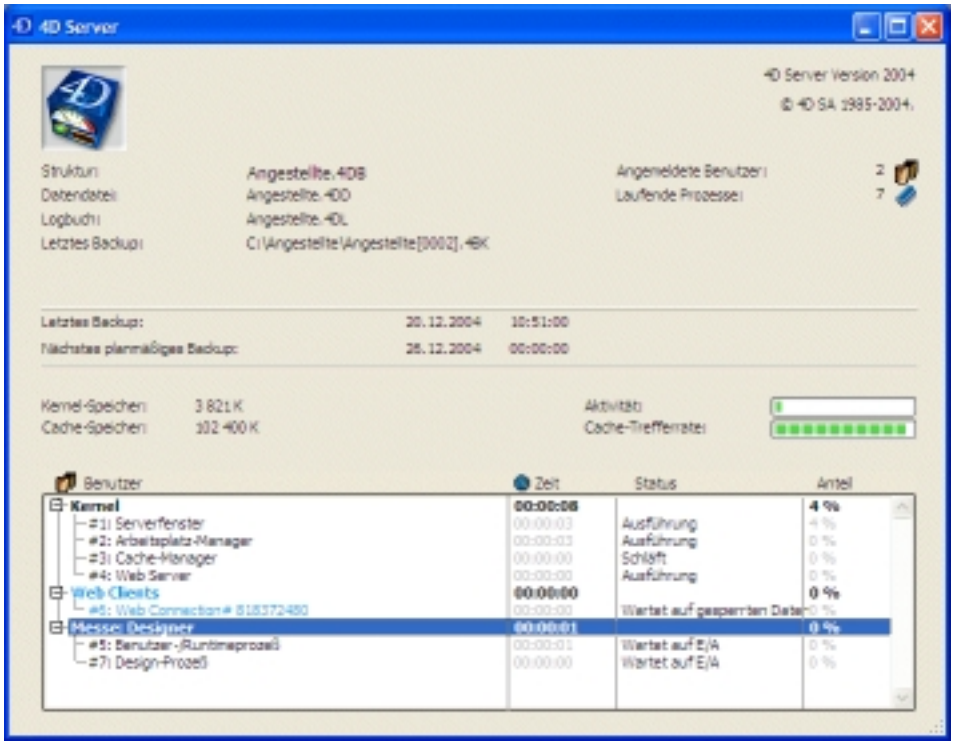
DatenbankOrdner/Preferences/Backup/Backup.xml

Dabei ist DatenbankOrdner der Ordner mit der Strukturdatei der Datenbank, die Unterordner Preferences/Backup/ werden bei Bedarf automatisch angelegt.

4

4D Server verwalten

Der Server wird über das Serverfenster grafisch verwaltet. Dieses Fenster zeigt Konfiguration und Aktivität der angemeldeten Rechner:



Das Serverfenster besteht aus zwei Bereichen:

• Der obere Teil enthält Informationen über die vom Server verwendeten Dateien, Backup und Speicher,

• Der untere Teil enthält Informationen über die angeschlossenen Benutzer und die laufenden Prozesse.

Allgemeine Informationen zum Server

Der obere Teil enthält im einzelnen:

• **4D Server Versionsnummer**: Versionsnummer Ihrer 4D Server Anwendung.

• **Strukturdatei**: Name der Strukturdatei, die 4D Server öffnet.

• **Datendatei**: Name der Datendatei, die der Strukturdatei zugeordnet ist.

• **Logbuch**: Name der Logbuchdatei zur fortlaufenden Protokollierung der Operationen in der Datenbank. Das Logbuch wird beim Erstellen der Datenbank mit angelegt, wenn es in den Einstellungen des Backup aktiviert ist.

• **Angemeldete Benutzer**: Anzahl der angemeldeten Benutzer.

• **Laufende Prozesse**: Anzahl der laufenden Prozesse. Dazu gehören alle Prozesse des Kernel, der angeschlossenen Benutzer, der Web Anbindungen und der Serverprozeduren.

• **Letztes Backup:** Zugriffspfad und Name der letzten Backup-Datei der Datenbank.

• **Letztes Backup:** Datum und Uhrzeit des zuletzt ausgeführten Backup.

• **Nächstes planmäßiges Backup:** Datum und Uhrzeit des nächsten automatischen Backup. Diese Information kommt von der automatischen Backup-Konfiguration, die in den Einstellungen der Datenbank definiert wird.

• **Kernel-Speicher**: Zugewiesener Speicher für den Kernel von 4D Server. Das Betriebssystem stellt diesen Speicher dynamisch ein je nach Einsatz der Datenbank und den vorhandenen Ressourcen.

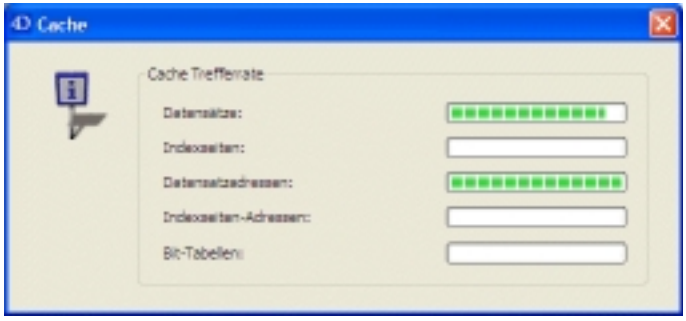
• **Cache-Speicher**: Zugewiesener Speicher für den Cache. Hier werden Daten zwischengelagert, um den Zugriff auf die Festplatte zu reduzieren. Diesen Speicher stellen Sie in den Einstellungen der Datenbank unter dem Thema Datenbank auf der Seite Datenbankverwaltung ein. Weitere Informationen dazu finden Sie im Kapitel "Einstellungen zur Konfiguration".

• **Ablaufbalken für die Serveraktivität**: Der Balken gibt den aktuellen Stand der Serveraktivität an. Je mehr Anfragen der Server erhält, desto weiter ist der Balken ausgefüllt.

•**Ablaufbalken für den Cache-Speicher**: Der Ablaufbalken zeigt an, wieviel Platz im Cache-Speicher vorhanden ist. Ist der Balken nur wenig ausgefüllt, sollten Sie die Speichergröße des Cache verringern.

Das Fenster Cache-Trefferrate

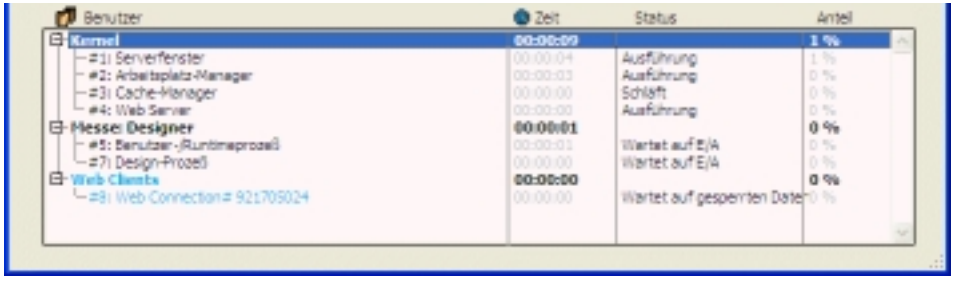
Klicken Sie den Balken der Cache-Trefferrate, sehen Sie die Elemente im Einzelnen:



Bei viel Speicherplatz können Sie jeder Anwendung den optimalen Cache-Speicherplatz zuteilen. Datensätze, Datensatzadressen und Indexseiten-Adressen werden in den Cache geladen, wenn Sie mit den Datensätzen arbeiten. Indexseiten und Bit-Tabellen werden geladen, wenn Sie Daten verändern, also Datensätze hinzufügen, ändern oder löschen. Im Beispiel finden Sie eine gängige Situation. Auch hier gilt: je mehr vom Ablaufbalken farbig ist, desto besser. Ist dagegen viel weiß, können Sie mit mehr Speicher die Arbeitsgeschwindigkeit verbessern. Bei größeren Datenbanken ist es normal, dass die Rubrik Datensätze nur halb gefüllt ist.
Weitere Hinweise zur Optimierung des Datenbestands finden Sie im Handbuch *4D_Tools*.

Informationen über Prozesse und Benutzer

Das **Prozess-Informationfenster** zeigt alle angeschlossenen Benutzer und die derzeit laufenden Prozesse an:



Alle **Benutzerprozesse** erscheinen in dieser Liste, davon ausgenommen sind die Prozesse, die lokal auf der Arbeitsstation laufen. Wollen Sie Prozesse ausblenden, klicken Sie unter Windows auf das Kreuz, auf Macintosh auf den Pfeil links neben dem Benutzernamen. Sind keine Benutzer angemeldet, erscheinen in diesem Fenster nur Prozesse der 4D Server Engine und optional Serverprozeduren.

Die 4D Server Engine wird von vier Kernel Prozessen gesteuert:

• **Serverfenster**: Dieser Prozess verwaltet die Benutzerschnittstelle des Servers. Sie antwortet auf Benutzeranfragen. Das können folgende Aktionen sein: Icon anklicken, Fenster auswählen oder verschieben. Dieser Prozess hat entweder den Status In Ausführung oder Schläft.

• **Arbeitsplatz-Manager:** Dieser Prozess verwaltet die Anmeldung einer neuen Arbeitsstation an den Server. Sein Status lautet standardmäßig "Wartet auf Eingabe/Ausgabe".

• **Cache-Manager:** Dieser Prozess verwaltet die Datenübertragung auf die Festplatte. Er ist nur aktiv, wenn der Cache voll ist bzw. eine Station den Befehl gegeben hat, den Cache zu sichern. Die Cache-Verwaltung sucht auch nach Anfragen für Indizes und aktiviert gegebenenfalls die Indexverwaltung.

• **Web Server**: Dieser Prozess verwaltet Web Anbindungen. Wenn Sie den Web Server Prozess starten, bedeutet das noch nicht, dass Sie eine aktuelle Web Anbindung öffnen. Sie ermöglichen damit lediglich, dass Web Benutzer Web Anbindungen starten können.

Jeder Prozess enthält folgende Angaben:

• Prozessname
• Zeitspanne (in Sekunden) die der Prozess seit dem Starten läuft.
• Prozessstatus
• Prozentsatz an Zeit, die 4D Server im Prozess benötigt.

Zur Unterscheidung werden die einzelnen Prozesse in verschiedenen Farben angezeigt:

- Kernel Prozesse in schwarz
- Benutzerprozesse in schwarz
- Prozesse für Web Anbindungen in cyan
- Prozesse für Serverprozeduren in blau
- Prozesse für Anbindungen mit 4D Open in grün

Hinweis: Abgebrochene Prozesse erscheinen für kurze Zeit in Rot.

Weitere Informationen finden Sie im Handbuch 4th Dimension Programmiersprache im Abschnitt Einführung in Prozesse.

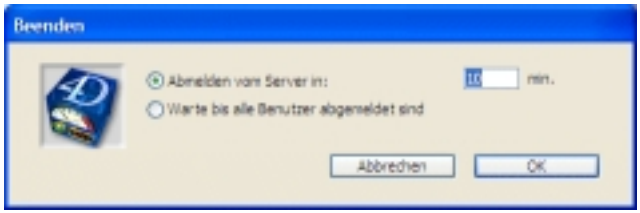
Referenz

Menü Prozesse.

Um 4D Server zu beenden, gehen Sie folgendermaßen vor:

1. Wählen Sie unter Windows im Menü **Datei** von 4D Server, auf MacOS im Menü **4D Server** der Anwendung den Befehl **Beenden**.

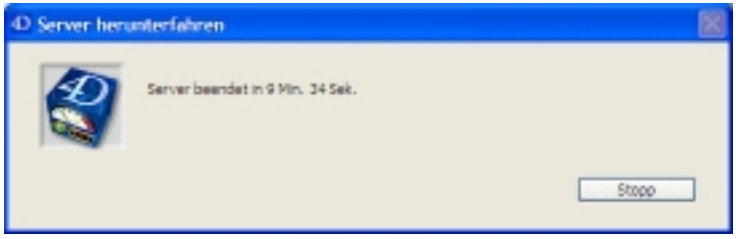
Auf dem Bildschirm des Server Rechners erscheint folgendes Dialogfenster:



2. Geben Sie entweder die Zeitspanne an, nach der Server beenden soll oder wählen Sie die Option **Warte, bis alle Benutzer abgemeldet sind**.

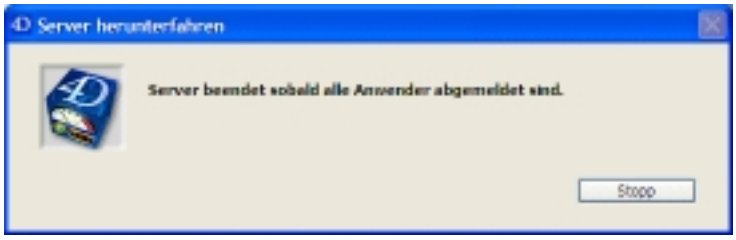
Sobald Sie eine Option gewählt haben, kann sich am Server kein neuer Client anmelden.

• Wählen Sie **Abmelden vom Server in ... min**, erscheint auf dem Bildschirm folgende Meldung:



Gleichzeitig erhalten alle Arbeitsstationen eine entsprechende Meldung. Der Benutzer kann die noch verbleibende Zeit weiterarbeiten:

• Wählen Sie **Warte, bis alle Benutzer abgemeldet sind**, erscheint folgendes Dialogfenster:



Auf der Arbeitsstation erscheint kein Dialogfenster. Der Server schaltet ab, sobald sich alle Arbeitsstationen abgemeldet haben.

Hinweise:

• In beiden Fällen beendet 4D Server sofort nach Bestätigen des Fensters, wenn keine Arbeitsstationen angemeldet sind.
• Soll 4D Server ohne Verzögerung beenden, tragen Sie als Zeitspanne 0 Minuten ein.
• Klicken Sie im Fenster "Server herunterfahren" auf die Schaltfläche **Abbrechen**, wird das Beenden des Servers nicht ausgeführt.

Unabsichtliches Beenden

Unterbricht eine Arbeitsstation die Verbindung zum Server nicht ordnungsgemäß, weil ein Netzwerkfehler aufgetreten ist bzw. die Arbeitsstation ausgeschaltet wurde, registriert 4D Server dies nach ca. 2 Minuten. Er führt dann automatisch folgende Operationen durch:

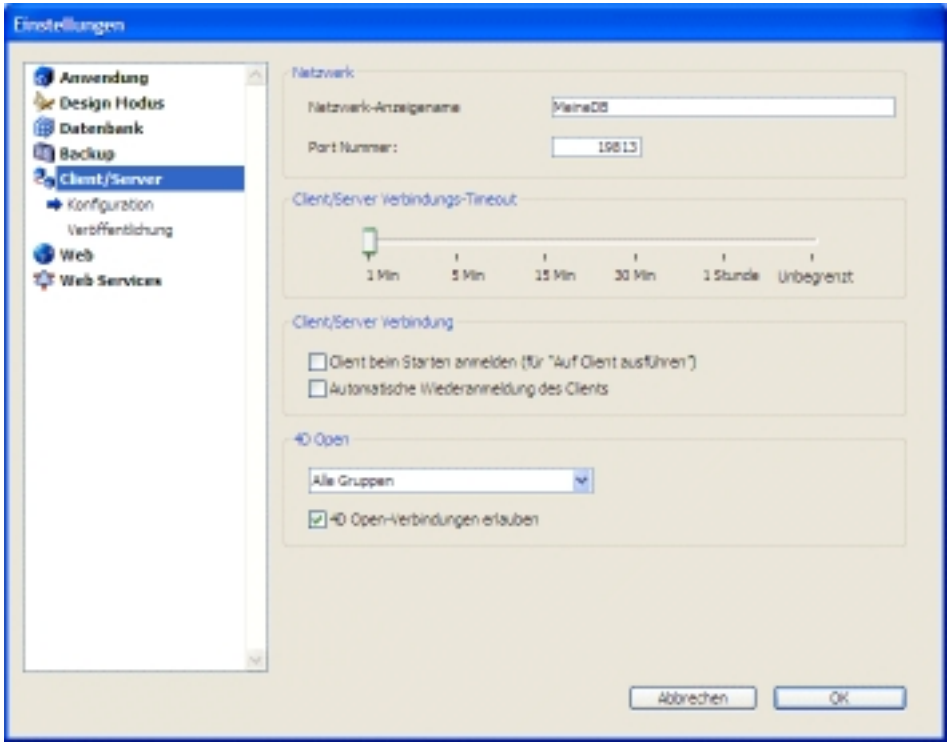
Er löscht die Prozesse der Arbeitsstation,
Er bricht laufende Transaktionen ab,
Er gibt evtl. gesperrte Datensätze frei,
Er gibt evtl. gesetzte Semaphoren frei.

Beendet der Server nicht ordnungsgemäß, erhalten die Arbeitsstationen die Meldung, dass die Verbindung zum Server unterbrochen wurde.

In den Einstellungen der Datenbank können Sie Parameter für das Netzwerk und die Client/Server Kommunikation festlegen (zugänglich für 4D Client und 4D Server), und bei Bedarf den Cache-Speicher selbst einstellen.

Konfiguration

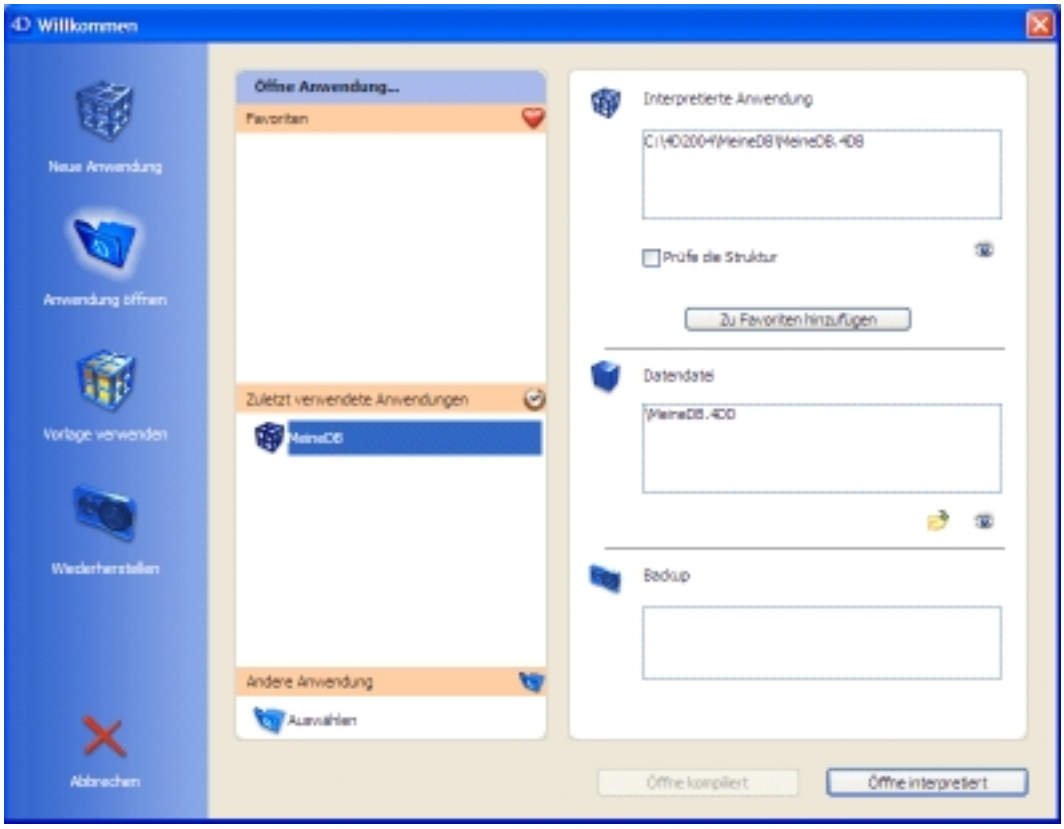
Das Dialogfenster **Client/Server** > **Konfiguration** enthält folgende Optionen:



Netzwerk

• **Netzwerk-Anzeigename:** Der Name einer Datenbank, die mit 4D Server veröffentlicht wird, lässt sich ändern. Das kann z.B. der Name auf der dynamischen Veröffentlichungsseite TCP/IP des Verbindungsdialogs sein.

4D Server verwendet standardmäßig den Namen der Strukturdatei der Datenbank. Sie können einen beliebigen Namen eingeben.



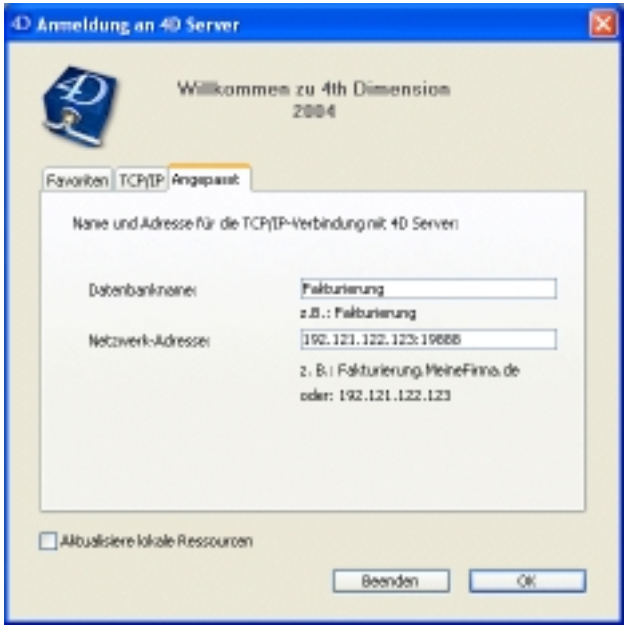
Hinweis: Dieser Parameter wird bei eigenen Client-/Server-Anwendungen nicht berücksichtigt. Im Prinzip meldet sich die Client-Anwendung direkt an die Server Anwendung an, ohne über den Anmeldedialog zu gehen. Der Dialog erscheint jedoch, wenn ein Fehler auftritt; in diesem Fall ist der Publikationsname der Server-Anwendung der Name der kompilierten Datenbank.

• **Port Nummer:** Mit dieser Option können Sie die TCP Port Nummer ändern, unter der 4D Server die Datenbank veröffentlicht. Diese Information wird in der Strukturdatei der Datenbank und auf jedem Client-Rechner gespeichert. 4D Server und 4D Client verwenden standardmäßig die Port Nummer 19813.
Sie müssen diesen Wert anpassen, wenn Sie mehrere 4D Anwendungen auf demselben Rechner mit dem TCP Protokoll verwenden wollen; in diesem Fall müssen Sie für jede Anwendung eine andere Port Nummer vergeben.

Ändern Sie diesen Wert auf 4D Server bzw. 4D Client, wird er automatisch auf alle 4D Client Rechner übertragen, die an die Datenbank angemeldet sind. Andere nicht angeschlossene Client-Rechner aktualisieren Sie, indem Sie bei der nächsten Anmeldung im Verbindungsdialog auf der Seite Angepasst die neue Port Nummer eingeben. Sie wird an die IP-Adresse des Server-Rechners angehängt, getrennt durch Doppelpunkt.

Hinweis: Ab 4D Server 2004 wird als Trenner der Doppelpunkt verwendet. Aus Kompatibilitätsgründen ist aber auch das Komma aus früheren Versionen zulässig.

Für die Nummer 19888 sieht das Dialogfenster folgendermaßen aus:



Hinweis: Auf der Seite **TCP/IP** zur dynamischen Veröffentlichung erscheinen nur die Datenbanken, die auf demselben Port, wie in 4D Client definiert, veröffentlicht werden.

Client-Server Verbindungs-Timeout

Im Ablaufbalken legen Sie das Timeout fest, d.h. die Dauer an Inaktivität, nach der die Verbindung zwischen 4D Server und den angeschlossenen Client-Rechnern beendet wird. Bei Unbegrenzt gibt es kein Timeout. In diesem Fall erfolgt keine Kontrolle der Client-Aktivität, d.h ein inaktiver Client bleibt auf unbestimmte Zeit an den Server angeschlossen.
Ist ein Timeout eingestellt, beendet der Server die Verbindung zu einem Client, wenn dieser innerhalb der vorgegebenen Zeitspanne nicht auf bestimmte Anfragen des Server reagiert.

Client beim Starten anmelden

Damit registriert 4D Server automatisch jeden 4D Client Rechner bei der Anmeldung an eine 4D Server Datenbank. Ein registrierter Client kann jede Operation ausführen, die vom Server oder einem anderen Client angefordert wird. Weitere Informationen dazu finden Sie im Abschnitt Clients registrieren.

Automatische Wiederanmeldung des Client

Mit dieser Option lassen sich die Client-Rechner in bestimmten Konfigurationen nach unerwarteter Abmeldung automatisch wiederanmelden. Diese Option ermöglicht eine stabilere Verbindung zwischen 4D Client und 4D Server, wenn im Netzwerk Probleme auftreten.
Ist diese Option markiert – sie gilt global für alle Clients – wird die Wiederanmeldung automatisch aktiviert, wenn ein Client die Verbindung zum Server verloren hat. Scheitert die Wiederanmeldung, wird der Standardfehler -10002 generiert. Ist sie erfolgreich, erscheint eine Meldung und unter Windows in der Task-Leiste das Symbol für 4D Client. Damit die automatische Wiederanmeldung ausführbar ist, muss das Timeout auf dem Server über 1 Minute liegen.

4D Open

Mit 4D Open, dem API (Application Programming Interface) können Benutzer über 4D bzw. „fremde“ Clients (4D for C/C++, ODBC-Treiber for 4D Server) auf 4D Server zugreifen. 4D Open lässt sich in 4th Dimension, 4D Server und 4D Client einbinden.

• **DropDown-Menü für 4D Open Zugriff:** Hier geben Sie die Benutzergruppe an, die über 4D Open eine Verbindung zur 4D Server Datenbank herstellen darf, wenn die Option “4D Open Verbindungen erlauben” markiert ist.

• **4D Open Verbindungen erlauben:** Mit dieser Option kann die in „4D Open Zugriff“ gewählte Gruppe über eine 4D Open Anwendung die Verbindung zu 4D Server herstellen.

Cache Speicher verwalten

In Version 2004 wurde die Verwaltung des Kernel-Hauptspeichers an die Möglichkeiten von Windows XP und MacOS X angeglichen. Der von 4th Dimension verwendete Speicher wird automatisch vom System zugewiesen und ist nicht länger anpassbar. Sie können jedoch weiterhin den Cache-Speicher Ihrer Datenbank anpassen.
Zum Setzen des Cache-Speichers gibt es zwei Berechnungsarten:

• Erzwungen
• Anpassbar

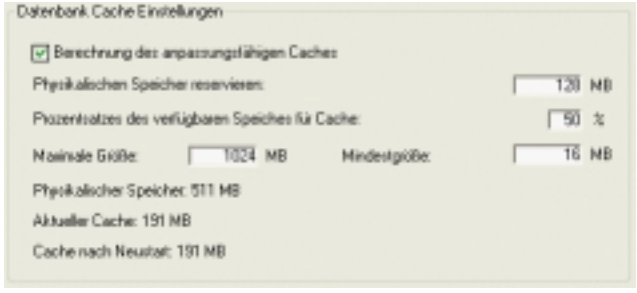
Erzwungen

In diesem Modus definieren Sie selbst die Größe des Cache-Speichers. Dazu genügt es, im Dialogfenster **Datenbank > Datenverwaltung** die Option "Berechnung des anpassungsfähigen Cache" zu deaktivieren. 4th Dimension zeigt dann einen Eingabebereich, in dem Sie selbst den Cache-Speicher festlegen, sowie Informationen zum tatsächlichen Speicher (auf dem Rechner verfügbarer RAM-Speicher), dem aktuellen Cache und dem Cache nach erneutem Hochfahren (berücksichtigt Ihre Änderungen) eintragen:

Die Größe des vorgegebenen Cache-Speichers wird für die 4D Datenbank reserviert, unabhängig von den für den Rechner verfügbaren Ressourcen. Diese Einstellung ist nur in wenigen Einzelfällen sinnvoll. In den meisten Fällen bietet der anpassbare Cache eine höhere Leistung.

Anpassbar

In diesem Modus wird der Cache-Speicher vom System dynamisch verwaltet – innerhalb der von Ihnen gesetzten Grenzen. Sie können einen leistungsstarken Cache-Speicher einrichten, der zu den meisten Konfigurationen passt. Dazu markieren Sie die Option Berechnung des anpassungsfähigen Cache. 4th Dimension zeigt dann mehrere Eingabebereiche:


Datenbank Cache Einstellungen
Berechnung des anpassungsfähigen Caches
Physikalischen Speicher reservieren: 128 MB
Prozentsatzes des verfügbaren Speichers für Cache: 50 %
Maximale Größe: 1024 MB
Mindestgröße: 16 MB
Physikalischer Speicher: 511 MB
Aktueller Cache: 191 MB
Cache nach Neustart: 191 MB

Die Größe des Cache-Speichers wird dann gemäß den Einstellungen dynamisch berechnet. Die standardmäßig vorgegebenen Werte entsprechen der standardmäßigen Verwendung von 4th Dimension.

• **Physikalischen Speicher reservieren:** Anteil des RAM-Speichers, der für das System und andere Programme zu reservieren ist. Dieser Wert wird zur Optimierung erhöht, wenn gleichzeitig mit 4th Dimension noch andere Programme auf dem Rechner laufen.

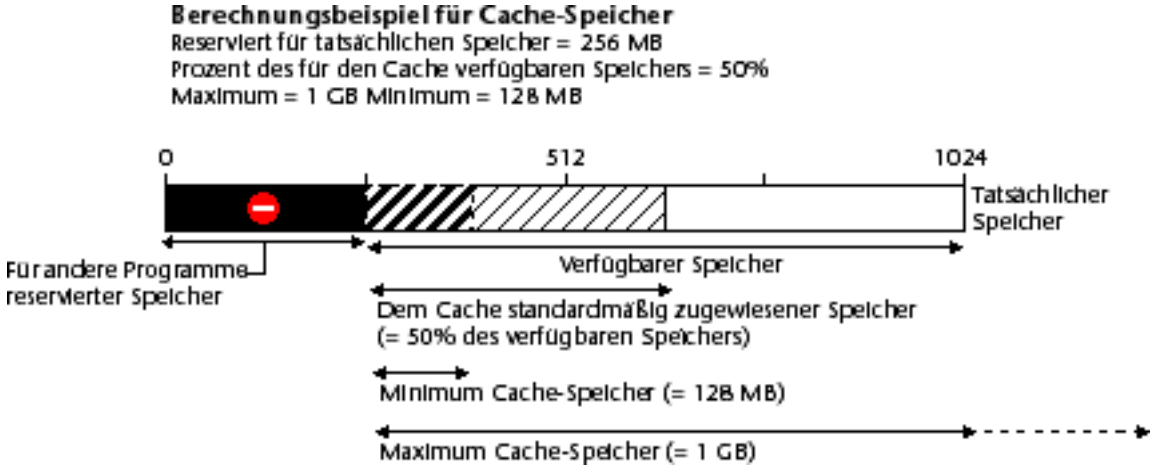
• **Prozentsatz des verfügbaren Speicher für Cache:** Prozent des verbleibenden Speichers, der standardmäßig dem Cache zugewiesen wird.
Um diese Größe zu bestimmen, führen Sie folgende Rechnung aus: (Tatsächlicher Speicher – Zu reservierender tatsächlicher Speicher) X Prozent des Speichers für den Cache.
Im anpassbaren Modus variiert die Größe des Cache-Speichers je nach den Erfordernissen der Programme und des Systems. Mit folgenden Optionen können Sie Einschränkungen setzen:

• **Maximale Größe:** Maximale Speichergröße, die der Cache verwendet. Der Wert kann max. 2000 MB (2 GB) sein.

• **Mindestgröße:** Mindestspeichergröße, die für den Cache reserviert werden muss. Der Wert kann nicht unter 4 MB sein.

Hinweis: Das Einrichten von Grenzen ist besonders hilfreich für Datenbanken, die auf Rechner verteilt werden, deren Speicherkonfiguration Sie nicht kennen. So können Sie in jedem Fall eine Mindestleistung gewährleisten.

Nachfolgendes Schema veranschaulicht die Speichereinteilung:


Berechnungsbeispiel für Cache-Speicher
Reserviert für tatsächlichen Speicher = 256 MB
Prozent des für den Cache verfügbaren Speichers = 50%
Maximum = 1 GB Minimum = 128 MB
0
512
1024
Tatsächlicher Speicher
Für andere Programme reservierter Speicher
Verfügbarer Speicher
Dem Cache standardmäßig zugewiesener Speicher (= 50% des verfügbaren Speichers)
Minimum Cache-Speicher (= 128 MB)
Maximum Cache-Speicher (= 1 GB)

Konfigurationsbeispiele

Nachfolgende Beispiele zeigen die Vorteile des anpassungsfähigen Cache mit unterschiedlicher Software und Speicherkonfiguration:

• Rechner mit 768 MB an RAM, auf dem ein einzelner 4D Server läuft.
In diesem Fall ist es sinnvoll, dem Server den Maximum Cache-Speicher zuzuweisen:
- Zu reservierender physikalischer Speicher = 256 MB
- Prozentsatz des verfügbaren Speichers für den Cache = 100%, da es nur eine 4D Server Anwendung gibt.
- Mindestgröße = 10 MB und Maximumgröße = 1 GB (in dieser Konfiguration sind diese Werte bedeutungslos)

Standardmäßige Cache-Größe: (768 - 256) x 100% = 512 MB

• Dieselbe Konfiguration wie oben, jedoch mit drei 4D Server Anwendungen. Jede soll denselben Cache erhalten.
Sie setzen den Wert für „Prozentsatz des verfügbaren Speichers für Cache“ für jeden Server auf 33%.

Standardmäßige Cache-Größe: (768 - 256) x 33% = 170 MB

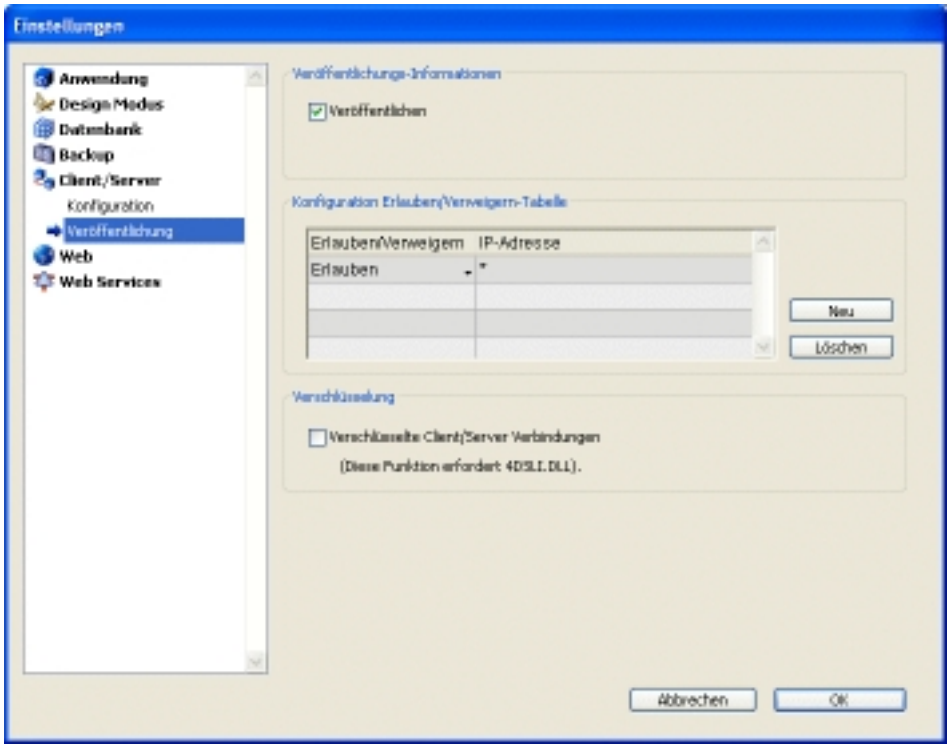
• Erstellung einer ausführbaren Anwendung für breitgefächerten Vertrieb mit multipler Konfiguration. Nach dem Testen ergibt sich, dass die Anwendung am besten mit einem Cache zwischen 20 und 100 MB operiert.
- Zu reservierender physikalischer Speicher = 256 MB
- Prozentsatz des verfügbaren Speichers für den Cache = 50%, damit Speicherplatz für andere Programme verfügbar bleibt.
- Mindestgröße = 20 MB und Maximumgröße = 100 MB.

Auf einem Rechner mit 256 MB an RAM verwendet 4th Dimension den Mindest Cache = 20 MB.
Auf einem Rechner mit 1 GB an RAM verwendet 4th Dimension den Maximum Cache = 100 MB.
Auf einem Rechner mit 384 MB an RAM verwendet 4th Dimension (384 - 256) x 50% = 64 MB.

Referenz

Clients registrieren, Einstellungen zur Veröffentlichung.

In den Einstellungen der Datenbank können Sie unter dem Thema **Client/Server** auf der Seite "Veröffentlichung" Parameter zur Sicherheit und zur Veröffentlichung von Datenbanken durch 4D Server festlegen (zugänglich für 4D Client und 4D Server):



Es gibt folgende Parameter.

Veröffentlichen

Hier definieren Sie, ob die 4D Server Datenbank in der Liste der veröffentlichten Datenbanken erscheint oder nicht:
• Ist die Option Veröffentlichen markiert, wird die Datenbank öffentlich gemacht und erscheint in der Liste der veröffentlichten Datenbanken. Diese Option ist standardmäßig aktiv.
• Ist die Option Veröffentlichen nicht markiert, wird die Datenbank nicht öffentlich gemacht. Sie erscheint nicht in der Liste der veröffentlichten Datenbanken (Seite **TCP/IP**). Zum Anmelden müssen Benutzer die Adresse der Datenbank im Verbindungsdialog auf der Seite **Angepasst** manuell eingeben.

Konfiguration Erlauben/Verweigern Tabelle

Mit dieser Tabelle setzen Sie die Regeln für den Zugriff auf die Datenbank. Das richtet sich nach den IP-Adressen des Client-Rechners. Das sorgt z.B. bei Anwendungen mit sensiblen Daten für mehr Sicherheit.

Hinweis: Diese Tabelle steuert nicht die Web Verbindungen.

Diese Tabelle funktioniert folgendermaßen:
• In der Spalte "Erlauben-Verweigern" wählen Sie im PopUp-Menü die anzuwendende Regel, d.h. **Erlauben** oder **Verweigern**. Um eine Regel hinzuzufügen, klicken Sie auf die Schaltfläche **Hinzufügen**. In der Tabelle erscheint dann eine neue Zeile. Über die Schaltfläche **Löschen** können Sie die aktuelle Zeile entfernen.
• In der Spalte "IP Adresse" setzen Sie die IP-Adresse(n), für welche die Regel gelten soll. Dazu klicken Sie in die Spalte und geben die Adresse in folgender Form ein: 123.45.67.89. Sie können auch nur den Anfang der Adresse eingeben, gefolgt von einem * (Stern): 192.168.* gibt alle Adressen an, die mit 192.168 beginnen.
• Die Anwendung der Regeln richtet sich nach der Reihenfolge der Anzeige in der Tabelle. Bei zwei widersprüchlichen Regeln hat die zuerst angelegte Regel Priorität.
Durch Ändern der aktuellen Sortierung können Sie die Zeilen neu ordnen (Klicken Sie auf den Kopfteil der Spalte, um die Sortierrichtung zu wechseln). Sie können die Zeilen auch per Drag&Drop bewegen.
• Aus Sicherheitsgründen ist die Anmeldung nur an Adressen möglich, die zu einer Regel passen, d.h. enthält die Tabelle eine bzw. mehrere Regeln Verweigern, werden alle Adressen abgewiesen, da keine zu mindestens einer Regel passt. Wollen Sie nur einige Adressen abweisen, andere jedoch zulassen, fügen Sie am Ende der Tabelle die Regel **Erlauben *** hinzu.

Beispiel:
Verweigere 192.168.* (weist alle Adressen ab, die mit 192.168 beginnen
Erlaube * (lässt jedoch alle anderen Adressen zu)

4D Server verwendet standardmäßig keine Verbindungseinschränkungen, d.h. die erste Zeile der Tabelle enthält die Bezeichnung **Erlaube** und das Zeichen * (alle Adressen sind zugelassen).

Verschlüsselte Client/Server Verbindungen

Mit dieser Option können Sie den SSL Modus (gesicherter Modus) zur Kommunikation zwischen dem Server-Rechner und den 4D Client-Rechnern ein- bzw. ausschalten. Weitere Informationen dazu finden Sie im Abschnitt Client/Server Verbindungen verschlüsseln.

Referenz

Client/Server Verbindungen verschlüsseln, Einstellungen zur Konfiguration.

Client/Server Verbindungen verschlüsseln

Sie können Client/Server-Verbindungen so einstellen, dass die 4D Clients von den Arbeitsstationen im gesicherten Modus mit 4D Server kommunizieren. Die gesicherte Client/Server Kommunikation basiert auf dem SSL Protokoll (*Secured Socket Layer*).

SSL Protokoll und Client/Server Verbindungen

Das SSL Protokoll sichert die Kommunikation zwischen zwei Anwendungen ab, das sind hauptsächlich Web Server und Browser. Das SSL Protokoll bürgt für die Identität von Sender und Empfänger, sowie für die Vertraulichkeit und Vollständigkeit der ausgetauschten Informationen. Eine ausführliche Beschreibung des SSL Protokolls finden Sie im Handbuch *4th Dimension Programmiersprache* im Kapitel Web Server im Abschnitt SSL Protokoll.

Mit dem SSL Protokoll können 4D Server und 4D Client mit größerer Sicherheit kommunizieren. 4D Server verwaltet die Operationen zum Erstellen des Schlüssels, der Authentifizierung und der Datenintegrität eigenständig. Der Benutzer muss keine weiteren Einstellungen vornehmen.

Hinweis: Das Verschlüsseln von Client und Server macht die Verbindung langsamer.

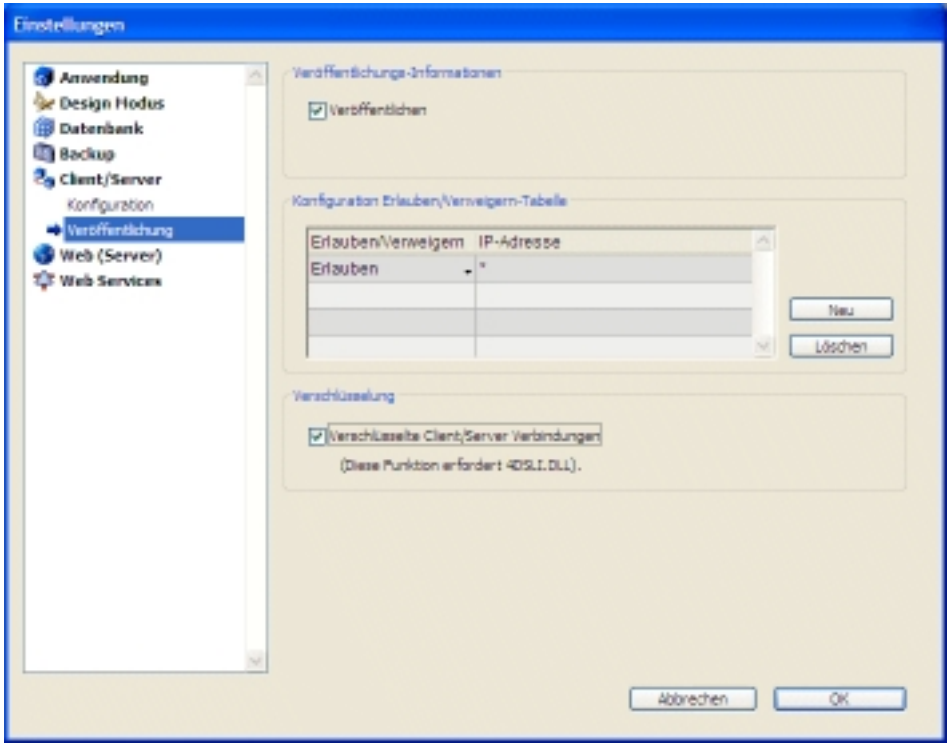
Einstellungen

Das SSL Protokoll wird in den Netzwerkkomponenten integriert und zwar zwischen dem TCP/IP Protokoll (niedere Ebene) und dem Protokoll auf hoher Ebene.

Um SSL in einer “klassischen” Client/Server-Architektur einzusetzen, achten Sie darauf, dass die Datei 4DSLI.DLL auf dem 4D Server Rechner sowie auf jedem 4D Client Rechner korrekt installiert sind. Das ist die Schnittstelle für die gesicherte Ebene (*Secured Layer Interface)* zum Verwalten von SSL. Diese Datei muss im Ordner [4D Extensions] der 4D Server-Anwendung/des Software-Pakets und der 4D Client Anwendungen/der Software-Pakete liegen, welche die Datenbank veröffentlichen.

Diese Elemente werden standardmäßig installiert.

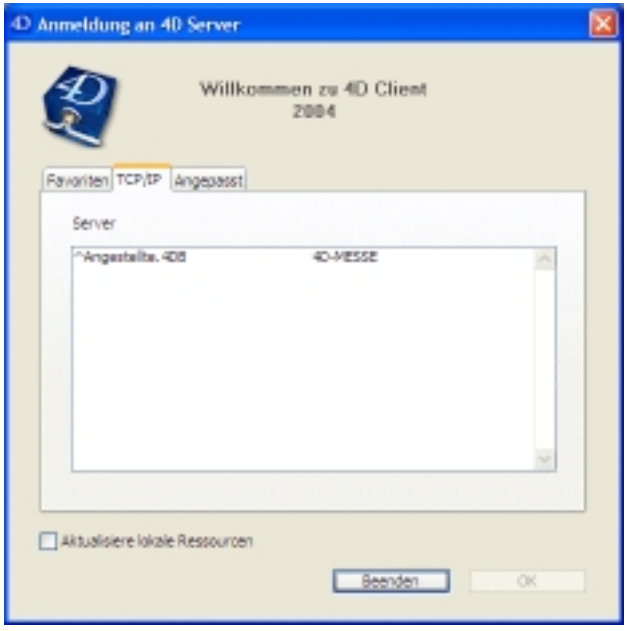
Dagegen ist die Verschlüsselung der 4D Server Verbindungen standardmäßig deaktiviert. Sie können diese in den Einstellungen der Datenbank unter dem Thema **Client/Server** auf der Seite “Veröffentlichung” aktivieren. Markieren Sie dazu im Bereich "Verschlüsselung" die Option **Verschlüsselte Client/Server Verbindungen**:



Diese Einstellung wird erst aktiv, wenn Sie 4D Server beenden und neu starten. Alle 4D Client Arbeitstationen melden sich dann im gesicherten Modus an.

4D Client Verbindung im gesicherten Modus

Eine im gesicherten Modus (SSL) veröffentlichte Datenbank ist im Verbindungsdialog auf der Seite TCP/IP mit einem Circumflex (^) gekennzeichnet:



Hinweis: Erscheint die Datenbank im Verbindungsdialog nicht automatisch, kann der Benutzer auf der Seite Angepaßt ihren Namen eingeben (siehe Abschnitt 4D Client anmelden und Einstellungen zur Veröffentlichung). In diesem Fall muss ein Circumflex (^) vor den Datenbanknamen gesetzt werden. Andernfalls wird die Verbindung zurückgewiesen.

Referenz

Keine Referenz.

5

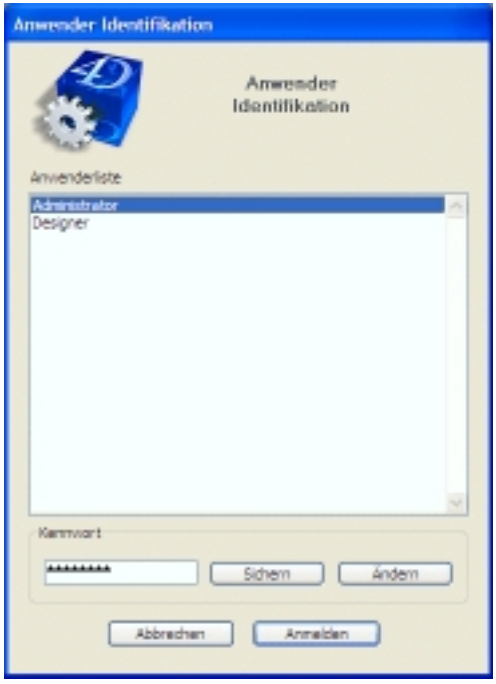
4D Client verwalten

Sie können Pfaddokumente anlegen, über die Sie vom Client-Rechner direkt auf eine 4D Server Datenbank zugreifen können.

Sie können das Pfad-Dokument per Drag&Drop oder Doppelklick auf die Datei auf die 4D Client Anwendung legen. Dann können Sie direkt auf die 4D Server Datenbank zugreifen, ohne Angaben im Anmeldedialog einzutragen. Dieser Dialog kann außerdem das Kennwort (verschlüsselt) enthalten, so dass Sie sofort Zugriff auf die 4D Server Datenbank haben.

Um ein Startdokument für eine Datenbank zu erstellen:

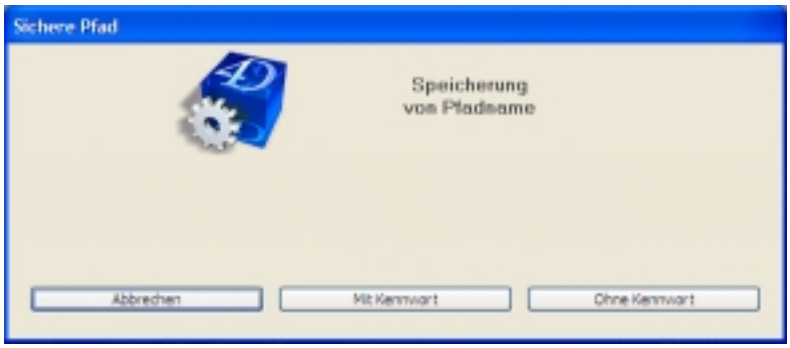
1. Geben Sie – je nach der aktuellen Konfiguration – im 4D Client Dialogfenster zur Identifikation den Benutzernamen ein oder wählen den Namen für die Zugriffsberechtigung aus und geben dann ein Kennwort ein.
Diese Parameter müssen gültig sein, damit der Zugriff auf das Dialogfenster zum Sichern der Verbindungsdatei möglich ist.



Hinweis: Dieses Dialogfenster erscheint unmittelbar, nachdem die 4D Server Datenbank ausgewählt ist, wenn das Kennwortsystem aktiviert ist.

2. Klicken Sie auf die Schaltfläche **Sichern**.

Auf dem Bildschirm erscheint folgendes Dialogfenster:

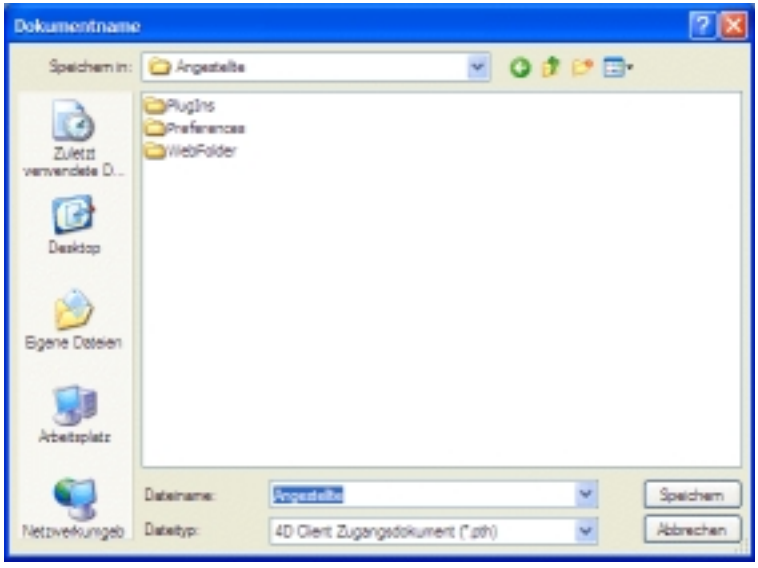


Sie können den Pfad mit oder ohne Kennwort speichern

• **Mit Kennwort:** In diesem Fall werden Zugriffspfad der Datenbank und Benutzerkennwort (verschlüsselt) im Startdokument gespeichert. Beim Starten von 4D Client erscheint kein Dialogfenster, die Verbindung erfolgt sofort.

• **Ohne Kennwort:** In diesem Fall wird der Zugriffspfad der Datenbank gespeichert, die Benutzer müssen jedoch ihr Kennwort angeben, um die Datenbank zu öffnen.

3. Klicken Sie auf die Schaltfläche **Mit Kennwort** oder **Ohne Kennwort**.
Auf dem Bildschirm erscheint ein Standard Sichern-Dialog, in dem Sie Name und Ort zum Speichern der Datei auf der Festplatte angeben. Der Name erhält die Endung ".pth".

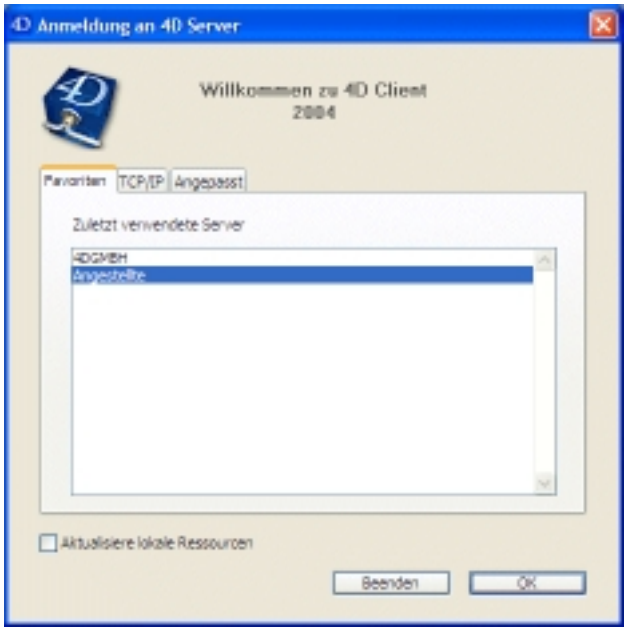


4. Wählen Sie Name und Ort für das Dokument und bestätigen Sie den Sichern-Dialog. Das Dokument der 4D Client Verbindung wird auf der Festplatte gespeichert.

Beim Starten von 4D Client erscheint ein Dialogfenster mit drei Registerkarten.

Registerkarte Favoriten

Die Seite Favoriten zeigt die Liste aller kürzlich verwendeten 4D Server:

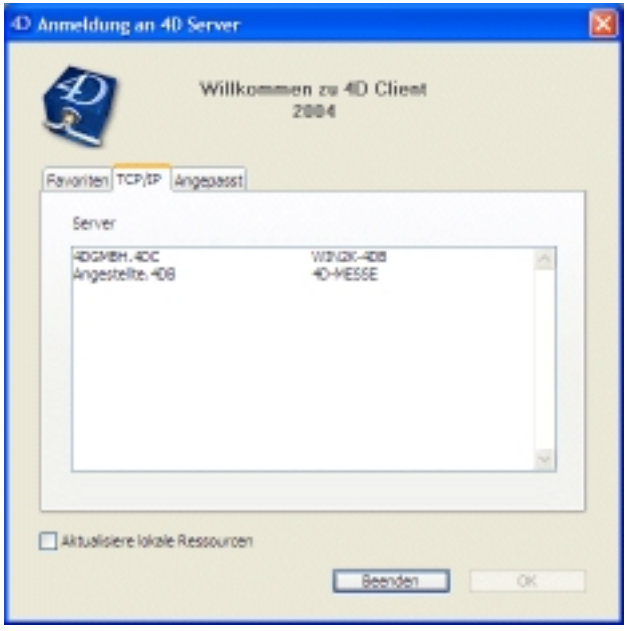


Die Liste ist alfabethisch sortiert. Um sich an einen Server in der Liste anzumelden, doppelklicken Sie auf seinen Namen oder markieren ihn und klicken auf die Schaltfläche **OK**.

Um einen Server aus der Liste zu entfernen, wählen Sie ihn aus und drücken die Taste **Del** bzw. die **Rückschrittaste**.

Registerkarte TCP/IP

Die Seite TCP/IP zeigt eine dynamische Liste aller 4D Server Datenbanken, die über das Netzwerk veröffentlicht werden:



Die Liste ist alfabethisch sortiert. Wollen Sie sich über diese Liste an einen Server anmelden, doppelklicken Sie auf seinen Namen oder markieren ihn und klicken auf die Schaltfläche **OK**.

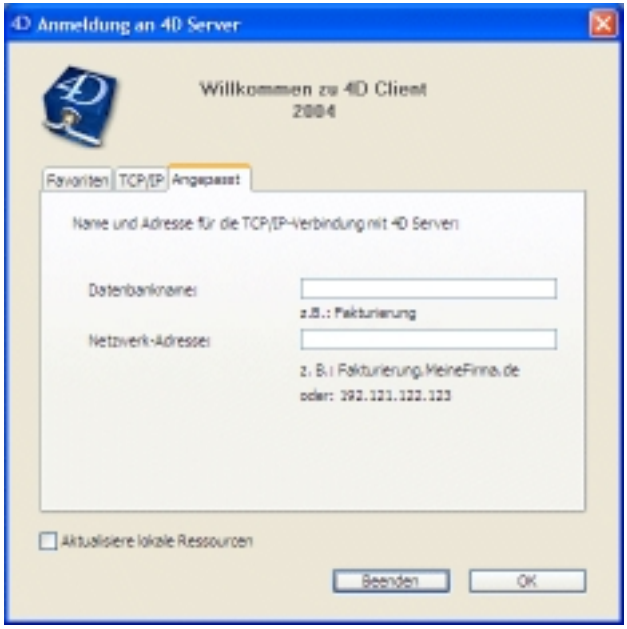
Hinweise:

• Ein Circumflex (^) vor dem Namen der Datenbanken gibt an, dass sie verschlüsselt publiziert werden. Weitere Informationen dazu finden Sie im Abschnitt Client/Server Verbindungen verschlüsseln.

• Die dynamische Veröffentlichtung des Datenbanknamen im Netzwerk lässt sich unterbinden. In diesem Fall muss die Verbindung manuell über die Seite "Angepasst" hergestellt werden. Weitere Informationen dazu finden Sie im Abschnitt Einstellungen zur Veröffentlichung.

Registerkarte Angepasst

Auf dieser Seite können Sie einen im Netzwerk veröffentlichten Server über seine IP Adresse zuweisen und einen eigenen Namen vergeben:



Sie können das TCP/IP Übertragungssystem von 4D Server so anpassen, dass die Namen der Datenbanken nicht automatisch über das Netzwerk publiziert werden. In diesem Fall erscheinen die Namen nicht auf der TCP/IP Seite des Clients. Kennen Sie jedoch die IP-Adresse einer nicht angezeigten Server-Datenbank, können Sie die Adresse manuell eingeben. Weitere Informationen dazu finden Sie im Abschnitt *Einstellungen zur Veröffentlichung*.

- **Datenbankname:** Hier definieren Sie den Namen der 4D Server Datenbank. Dieser Name wird auf der Seite **Favoriten** verwendet, wenn Sie Bezug auf die Datenbank nehmen.
- **Netzwerk-Adresse:** Hier geben Sie die IP-Adresse des Rechners an, auf dem der 4D Server gestartet wurde. Laufen auf einem Rechner zwei Server gleichzeitig, müssen auf die IP Adresse ein Doppelpunkt und eine Port-Nummer folgen, zum Beispiel 192.168.92.104:19814.

Hinweis: Ab 4D Server 2004 wird als Trenner der Doppelpunkt verwendet. Aus Kompatibilitätsgründen ist aber auch das Komma aus früheren Versionen zulässig.

Der Port zum Veröffentlichen eines Servers ist standardmäßig 19813. Diese Nummer lässt sich in den Einstellungen der Anwendung verändern. Weitere Informationen dazu finden Sie im Abschnitt *Einstellungen zur Konfiguration*.

Hinweis: Haben Sie auf den Seiten **Favoriten** oder **TCP/IP** eine Datenbank ausgewählt und klicken auf die Registerkarte **Angepaßt**, enthalten die beiden Felder diese Informationen.

Ist auf dieser Seite ein Server zugewiesen, können Sie sich durch Anklicken der Schaltfläche **OK** an den Server anmelden. Der Server erscheint dann auf der Seite Angepasst.

Hinweis: Wird die Datenbank verschlüsselt veröffentlicht, müssen Sie vor den Namen ein Circumflex (^) setzen. Andernfalls wird die Verbindung zurückgewiesen. Weitere Informationen dazu finden Sie im Abschnitt Client/Server Verbindungen verschlüsseln.

Lokale Ressourcen aktualisieren

Mit dieser Option können Sie die Aktualisierung der lokalen Ressourcen auf dem Client-Rechner beim Anmelden erzwingen. Das sind Informationen zur Struktur der Datenbank, die auf jedem Client-Rechner gespeichert sind.
In der Regel werden die lokalen Ressourcen automatisch aktualisiert, wenn die Struktur der Datenbank zwischen zwei Anmeldungen geändert wurde. In den meisten Fällen ist diese Option unnötig. In einigen spezifischen Fällen kann es jedoch erforderlich sein, die Aktualisierung zu erzwingen.

Ordner Extras

In der Client-/Server Konfiguration ist es möglich, einen spezifischen Ordner zum Übertragen eigener Elemente des Servers auf Client-Rechner zu verwenden (Ressourcendatei, Textdokumente, Dateien mit XML Einstellungen, etc.). Dieser Ordner mit Namen **Extras** muss neben die Strukturdatei gelegt werden.

4D Server verwaltet automatisch die Änderungen an dieser Datei und überträgt nur das, was notwendig ist. Außerdem wird der Inhalt des Ordners komprimiert, um die Übertragungszeit über das Netzwerk zu optimieren.

Wichtig: Vermeiden Sie die Übertragung großer Datenmengen, da dies die Dauer der Aufbauphase der Verbindung für Client Rechner signifikant erhöhen kann.

Auf der Client-Seite wird der Ordner **Extras** an dieselbe Stelle wie die anderen Strukturelemente gelegt, z.B.:

System	Pfad für die auf den Client-Rechner geladenen Ordner
Mac OS	{Disk}:Library:Application Support:4D:DatabaseName_Address:Extras
Windows 2000 Windows XP	{Disk}:\Documents and Settings\User Name\Application Data\ 4D\DatabaseName_Address\Extras

Die ursprüngliche Hierarchie wird auf jedem Client-Rechner rekonstruiert.

Referenz

Keine Referenz.

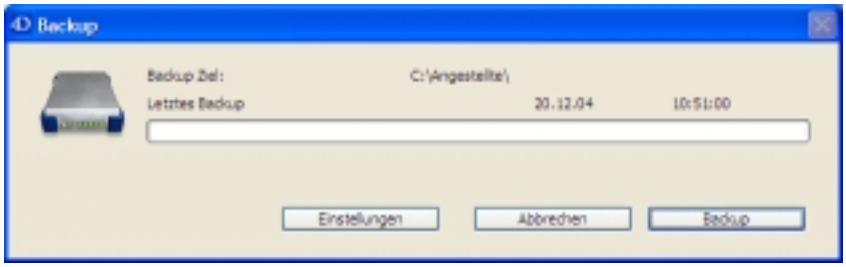
6

4D Server Menüs

Das Menü **Datei/Ablage** enthält folgende Befehle:

Backup

Mit diesem Befehl können Sie jederzeit ein Backup der Datenbank starten. Wählen Sie diesen Befehl, erscheint folgendes Dialogfenster:



• Die Schaltfläche **Backup** startet sofort ein Backup mit den Parametern, die in den Einstellungen der Datenbank der Anwendung definiert wurden (Dateien für das Backup, Ort der Archive, Anzahl der Backup-Sätze)

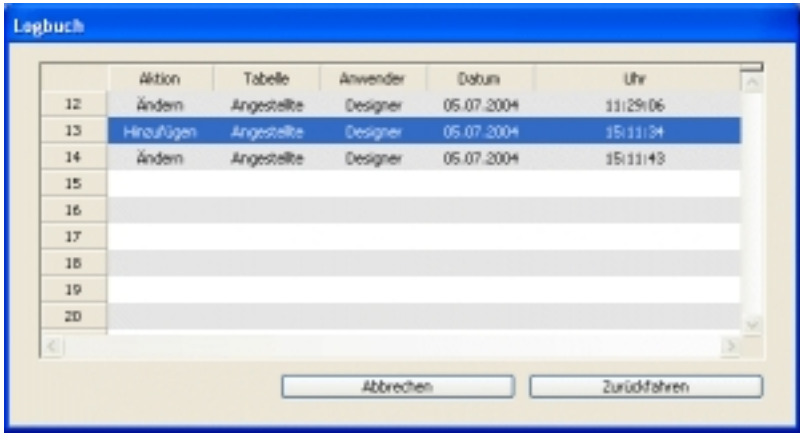
• Die Schaltfläche **Einstellungen** öffnet die Einstellungen der Datenbank unter dem Thema **Backup**. Hier können Sie die aktuellen Einstellungen für das Backup ansehen und bei Bedarf verändern.

• Die Schaltfläche **Abbrechen** unterbricht die Operation der Datensicherung.

Prüfe Logbuch

Erstellen Sie eine neue Datenbank, können Sie gleichzeitig die zugehörige Logbuchdatei anlegen. Eine Logbuchdatei ist eine Art Journal. Sie verzeichnet alle Änderungen in der Datenbank, wie z.B. Hinzufügen, Ändern und Löschen von Dateien, Datenfeldern bzw. Datensätzen – egal, ob Sie von einem Benutzer auf der Arbeitsstation, einer Serverprozedur, einem Plug-In, einem Web Browser oder auch über 4D Open ausgeführt werden.

Bei einem Störfall in der Datenbank sorgen das Wiederherstellen des letzten Backup und die Integration des Logbuchs dafür, dass die Datenbank exakt in dem Stadium vor dem Störfall wiederhergestellt wird.
Der Befehl **Logbuch:Rollback** öffnet ein Fenster mit dem Inhalt des aktuellen Logbuchs:



Dieses Fenster ist hilfreich, um die in der Datenbank ausgeführten Operationen zu analysieren und den Auslöser für Fehler oder Störungen herauszufinden. Sie können insbesondere die Operationen auf jedem Client-Rechner prüfen.
Über die Schaltfläche **Zurückfahren** können Sie in den ausgeführten Operationen rückwärts blättern.

Als Service registrieren
Registrierung für aktuelle Datenbank aufheben
Alle 4D Server Registrierungen aufheben

Sie können 4D Server als Service unter Windows und auf MacOS starten.

Um eine 4D Server Datenbank als Service zu registrieren, wählen Sie im Menü **4D Server** den Befehl **Als Service registrieren**. Beim nächsten Start des Rechners startet 4D Server automatisch und öffnet die aktuelle Datenbank.

Sie können beliebig viele Datenbanken registrieren. Jede Datenbank lässt sich nur einmal registrieren.

Warnung: Vergewissern Sie sich, dass Sie beim Öffnen der Sitzung ein gültiges Benutzerkonto verwenden. Es muss auch auf einen Drucker zugreifen können, sonst erscheint eine Fehlermeldung. Der Zugriff auf die Einstellungen von Services erfolgt in der Regel über Verwaltungs-Tools/Services der Kontrollfelder.

Um die Registrierung einer Datenbank aufzuheben, wählen Sie im Menü **4D Server** den Befehl **Registrierung für aktuelle DB aufheben**.

Um die Registrierung aller Datenbanken aufzuheben, wählen Sie im Menü **4D Server** den Befehl **Alle 4D Server Registrierungen aufheben**. Er erscheint in Grauschrift, wenn die Datenbank nicht als Service registriert ist.

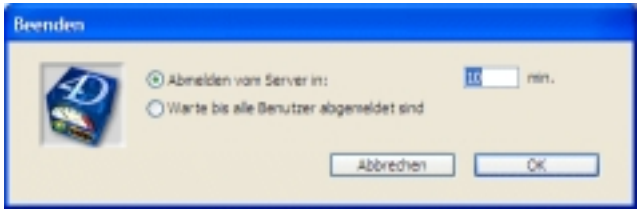
Startet die Anwendung beim Öffnen als Service, können Sie diesen Status nicht in 4D Server ändern; die drei Menübefehle sind inaktiv. Um den Service zu stoppen, rufen Sie das Kontrollfeld Dienste auf.

Beenden

Um den Server abzuschalten, gehen Sie folgendermaßen vor:

Wählen Sie unter Windows im Menü **Datei**, auf MacOS im Menü **4D Server** der Anwendung den Befehl **Beenden**.

Auf dem Bildschirm erscheint folgendes Dialogfenster



Abmelden vom Server in ... min: Geben Sie hier die Zeitspanne an, nach der sich Server und alle Arbeitsstationen automatisch abschalten sollen.

Auf dem Bildschirm erscheint folgendes Dialogfenster:

Auf der Arbeitsstation erscheint eine entsprechende Meldung. Sie können die noch verbleibende Zeit weiterarbeiten.

Warte, bis alle Benutzer angemeldet sind: Der Server beendet, sobald alle Anwender beendet haben. Hat ein Anwender nicht rechtzeitig beendet, wird die Verbindung trotzdem unterbrochen. Neue Anwender können sich nicht mehr anmelden, da der Server nicht mehr im Netzwerk erscheint.

Unabsichtliches Beenden

Unterbricht eine Arbeitsstation die Verbindung zum Server nicht ordnungsgemäß, weil ein Netzwerkfehler aufgetreten ist bzw. die Arbeitsstation ausgeschaltet wurde, registriert 4D Server dies nach ca. 2 Minuten. Er führt dann automatisch folgende Operationen durch:

Er löscht die Prozesse der Arbeitsstation,
Er bricht laufende Transaktionen ab,
Er gibt evtl. gesperrte Datensätze frei,
Er gibt evtl. gesetzte Semaphoren frei.

Beendet der Server nicht ordnungsgemäß, erhalten die Arbeitsstationen die Meldung, dass die Verbindung zum Server unterbrochen wurde.

Referenz

Backup verwalten, Logbuch verwalten.

Das Menü **Bearbeiten** von 4D Server enthält die Standardbefehle Kopieren/Einfügen, den Befehl **Zeige Zwischenablage**, etc.

Unter Windows enthält dieses Menü auch den Befehl **Einstellungen**, der das Dialogfenster Einstellungen der Anwendung anzeigt. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Designmodus*.

Hinweis: Auf MacOS liegt der Befehl **Einstellungen** im Menü **4D Server** der Anwendung.

Referenz

Menü Datei/Ablage, Menü Daten, Menü Hilfe, Menü Prozesse, Menü Web Server.

Prozesse im Serverfenster steuern Sie über das Menü **Prozesse**.

Löschen

Warnung: Dieser Befehl dient zur Verwaltung und zur Fehlerbehebung. Sie sollten ihn nur verwenden, wenn ein Client einen Vorgang nicht selbst beenden kann.

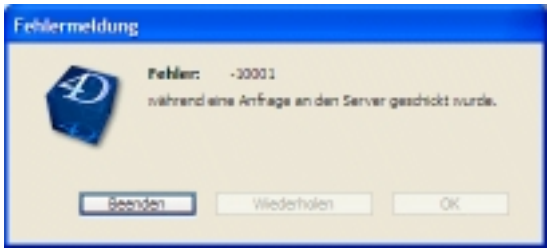
Der Befehl **Löschen** hat je nach dem ausgewählten Prozesstyp eine andere Wirkung:

• Wählen Sie den Prozess für eine Verbindung aus, wird dieser Prozess abgebrochen.
• Wählen Sie einen Benutzer aus, werden alle Prozesse abgebrochen, die zu diesem Benutzer gehören. Der Benutzer wird aus der Liste entfernt und automatisch vom Server abgemeldet.
• Wählen Sie eine Serverprozedur aus, wird diese Serverprozedur abgebrochen.
• Wählen Sie die Liste der Serverprozeduren aus, werden alle Serverprozeduren abgebrochen.
• Wählen Sie den Prozess für eine Web-Anbindung aus, wird dieser Prozess abgebrochen.
• Wählen Sie die Liste der Web Clients aus, werden alle Prozesse der Web Anbindungen gelöscht.

Sie können alle Prozesse außer Kernelprozesse löschen. Ist ein Kernel-Prozess ausgewählt, erscheint der Befehl in Grauschrift, d.h. er ist deaktiviert.

Versuchen Sie, einen abgebrochenen Prozess auf einem Client-Rechner zu verwenden, erscheint die Meldung, dass der Prozess nicht länger verfügbar ist.

Beispielsweise erscheint folgende Meldung, wenn der Befehl **Löschen** auf einen Benutzer angewendet wird:



Fehler -10001, Die aktuelle Verbindung zur Datenbank wurde unterbrochen.

Hinweis: Eine Liste der Netzwerkfehler finden Sie im Handbuch *4th Dimension Programmiersprache* im Abschnitt Netzwerkfehler.

Beendet eine Arbeitsstation die Verbindung zum Server unabsichtlich, erkennt 4D Server dies nach ein paar Sekunden. Er führt dann folgendes aus:

- Er hebt die Sperre für alle gesperrten Datensätze auf
- Er annulliert alle noch nicht beendeten oder bestätigten Transaktionen
- Er annulliert alle Client Prozesse
- Er entfernt den Benutzer im 4D Server Prozessfenster aus der Liste der Benutzer

Wollen Sie nicht so lange warten, können Sie den Prozess über den Befehl **Löschen** selbst beenden.

Warten zwei Arbeitsstationen aufeinander, und kann keine von beiden die Situation lösen, wenden Sie den Befehl **Löschen** an, um diese Deadlock-Situation zu beenden.

Erweitern/Übersicht

Wählen Sie den Befehl **Erweitern**, sehen Sie die Prozesse des Servers und der angeschlossenen Arbeitsstationen.

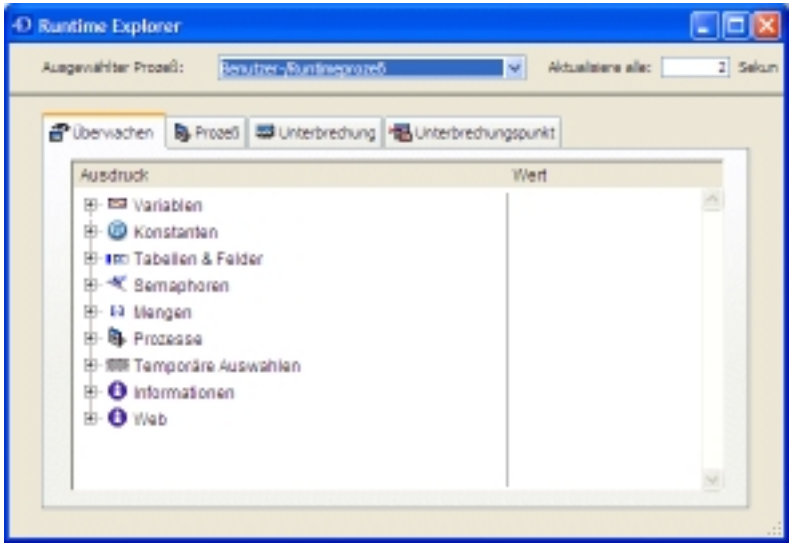
Wählen Sie den Befehl **Übersicht**, können Sie die Prozesse ausblenden. Sie enthält nur noch die Kernelreferenz und die Arbeitsstationen. Haben Sie zuvor den Kernel bzw. eine Arbeitsstation ausgewählt, gelten diese Befehle nur für das ausgewählte Objekt. Sie können mit Ausnahme der Kernelprozesse sämtliche Prozesse ganz oder teilweise ausblenden.

Prozessfenster einblenden/ausblenden

Das Prozessfenster ist standardmäßig eingeblendet. Mit dem Befehl **Prozessfenster ausblenden** können Sie es ausblenden.

Runtime Explorer einblenden/ausblenden

Mit diesem Befehl können Sie das Fenster 4D Server Runtime Explorer ein- oder ausblenden.



Mit dem Runtime Explorer können Sie den Status verschiedener Strukturelemente der Datenbank ansehen und prüfen, ob die verfügbaren Ressourcen korrekt verwaltet werden. Der Runtime Explorer ist besonders hilfreich beim Entwickeln oder Analysieren einer Datenbank.

Das Fenster enthält vier Registerkarten, über die Sie auf die Seiten Überwachen, Prozess, Unterbrechung und Unterbrechungspunkt gelangen. Er funktioniert in 4D Server auf dieselbe Art und Weise wie in 4th Dimension. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Designmodus.*

Schrittmodus

Der Befehl **Schrittmodus** kann für folgende Prozesse angewandt werden:

• Prozess für Verbindung, um Trigger im Schrittmodus zu durchlaufen, die innerhalb dieses Prozesses ausgeführt werden.

• Prozess für Web Anbindung, um Code im Schrittmodus zu durchlaufen, der als Antwort auf Anfragen, die der Web Browser sendet, auf dem Server-Rechner ausgeführt wird.

• Serverprozedur, um die Projektmethode, die als Serverprozedur auf dem Server-Rechner ausgeführt wird, im Schrittmodus zu durchlaufen.

Wählen Sie den Befehl **Schrittmodus**, erscheint das Fenster Debugger für den ausgewählten Prozess, sobald dieser Prozess Programmiercode ausführt. Das Fenster Debugger für einen Prozess erscheint auch, wenn Sie den Befehl **Schrittmodus** in einer Methode wählen, die diesen Prozess ausführt.

Interessant dabei ist, dass sich 4D Server die Schrittanfrage merkt:

• Führt der Prozess gerade Programmiercode aus, erscheint sofort der Debugger für diesen Prozess.
• Führt der Prozess gerade keinen Programmiercode aus, weil er z.B. auf ein Ereignis im Eingabemodus wartet, erscheint der Debugger unmittelbar, nachdem der Prozess die Ausführung des Codes abgeschlossen hat.

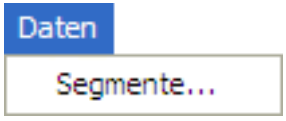
Hinweis: Dies ist eine Funktion für Entwickler, die mit 4D Server im Team arbeiten. Bei kompilierten Anwendungen lässt sich der Debugger nicht aufrufen.

Weitere Informationen zum Schrittmodus finden Sie im Handbuch *4th Dimension Programmiersprache* unter dem Befehl TRACE.

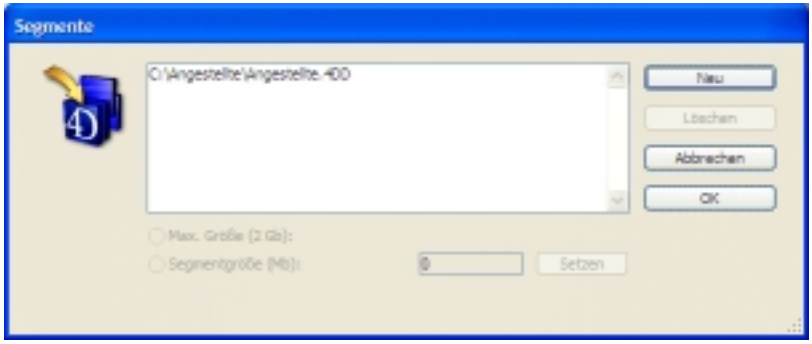
Referenz

Menü Bearbeiten, Menü Datei/Ablage, Menü Daten, Menü Hilfe, Menü Web Server.

4D Server ermöglicht Ihnen das Erzeugen von bis zu 128 Gigabyte großen Datenbeständern, unterteilt in maximal 64 Segmente von jeweils 2 Gigabyte. Um die Größe der Datei über 2 Gigabyte hinaus zu erhöhen, fügen Sie Datensegmente hinzu. Dies ermöglicht Ihnen, verschiedene Segmente einer Datei auf verschiedenen Volumes abzulegen.



Wählen Sie im Menü **Daten** den Befehl **Segmente.** Auf dem Bildschirm erscheint folgendes Dialogfenster:



Sie können eine Datei segmentieren, wenn Sie die Datenbank erzeugen oder nachdem Sie angefangen haben, sie zu benutzen. Sie können eine neue Datei segmentieren, wenn Sie erwarten, dass die Datei sehr groß wird. Das Segmentieren einer Datei ermöglicht die Speicherung einer virtuell unbegrenzten Anzahl von Daten.

Hinweis: Sie müssen keine Datensegmente erzeugen, wenn Sie weniger als 2 Gigabyte Daten haben oder Ihre Festplatte die voraussichtliche Größe Ihrer Datei speichern kann.

Beim Segmentieren einer Datei unterteilen Sie die Datei in Segmente und legen dann fest, auf welchen Volumes das jeweilige Segment gespeichert wird. Zum Beispiel können 4 Gigabyte Daten in zwei Segmente von jeweils 2 Gigabyte unterteilt werden. Jedes Segment kann in der Größe begrenzt sein, so dass Sie Platz auf Ihrer Festplatte für andere Dateien reservieren können und ein vollkommen volles Volume vermeiden.

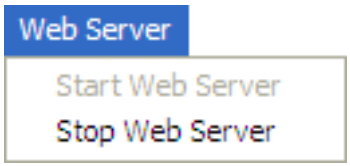
4D Server füllt die Datensegmente transparent in der Reihenfolge, in der sie erzeugt wurden. Wenn ein Segment voll ist, geht das Programm automatisch zum nächsten über. Wenn Sie durch Löschen von Daten Platz in einem Datensegment schaffen, werden die Lücken im Segment wiederverwendet. Wenn alle Datensegmente voll sind, erhalten Sie eine Meldung, dass die Volumes, auf denen Segmente vorhanden sind, voll sind. Dann können Sie weitere Datensegmente hinzufügen.

Hinweis: Wird das Logbuch auf einer anderen Festplatte angelegt, erhalten Sie auch eine Fehlermeldung, wenn dieses Medium voll ist.

Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Designmodus*.

Referenz

ADD DATA SEGMENT, Menü Bearbeiten, Menü Datei/Ablage, Menü Hilfe, Menü Prozesse, Menü Web Server.



Über dieses Menü können Sie den Web Server jederzeit starten und beenden.

Hinweis: Auf MacOS funktioniert dieses Menü evtl. nicht, wenn der Web Server über eine temporäre Root-Sitzung gestartet wird. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* im Abschnitt *Web Server konfigurieren und Verbindung verwalten*.

Es gibt drei verschiedene Möglichkeiten, den 4D Web Server zu starten:

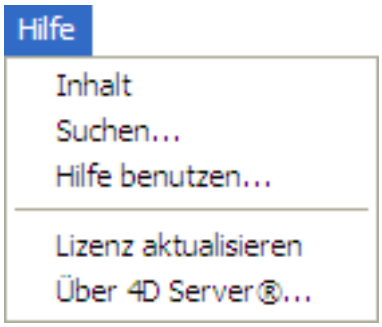
• Web Server selbst aufrufen
Wählen Sie in 4D Server das Menü **Web Server**, in 4D Client das Menü **Start**. Mit den beiden Befehlen **Start Web Server** und **Stop Web Server** können Sie den Web Server beliebig an- und ausschalten.

• Die Datenbank wird beim Öffnen automatisch im Web publiziert.
Dazu müssen Sie in den Einstellungen der Anwendung auf der Seite Web>Konfiguration unter den Web Server Start Optionen "Automatisch starten" markieren. Öffnen Sie nun Ihre Datenbank mit 4th Dimension oder 4D Server, wird sie automatisch im Web publiziert.

• Über das Programm durch Aufrufen des Befehls START WEB SERVER.

Referenz

Menü Bearbeiten, Menü Datei/Ablage, Menü Daten, Menü Hilfe, Menü Prozesse.

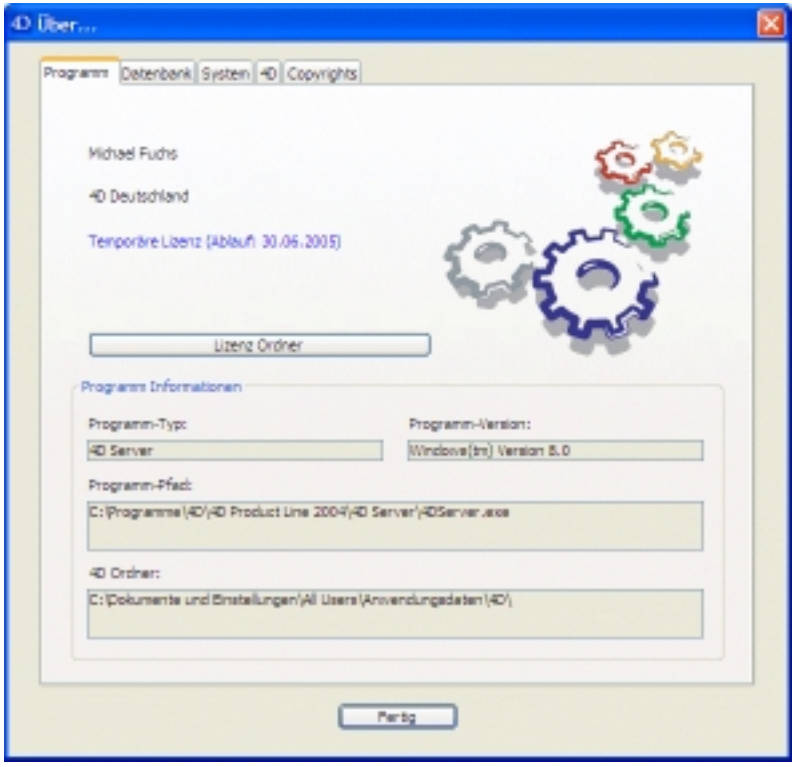


Mit den ersten Befehlen im Menü **Hilfe** können Sie auf die Online-Hilfe für 4D Server zugreifen.

Lizenz aktualisieren

Dieser Befehl öffnet das Dialogfenster, um zusätzliche Lizenzen in Ihrer 4D Umgebung zu aktivieren. Weitere Informationen dazu finden Sie im *Installationshandbuch 4D Produkte*.

Der Befehl **Über 4D Server** zeigt ein Dialogfenster mit Registerkarten, über die Sie Informationen über das Programm 4D Server (Name und Plazierung der Dateien), die laufende Datenbank und Systemdateien erhalten:



Über die Seite **4D** können Sie auf die Online-Registrierung von 4D Server zugreifen.

Hinweis: Auf MacOS liegt der Befehl **Über 4D Server** im Menü **4D Server** der Anwendung.

Referenz

Menü Bearbeiten, Menü Datei/Ablage, Menü Daten, Menü Prozesse, Menü Web Server.

7

4D Programmiersprache

4D Server bietet drei Möglichkeiten, den 4D Code auf dem Server-Rechner auszuführen:

- Trigger
- Serverprozeduren
- Datenbankmethoden

Trigger

Trigger sind Tabellenereignisse, die ablaufen, wenn bestimmte Ereignisse auftreten, die das Sichern, Löschen und Laden von Datensätzen betreffen. Sie unterbinden unzulässige Operationen in den Datensätzen und schützen vor unabsichtlichem Löschen. In einem Rechnungssystem kann ein Trigger zum Beispiel dafür sorgen, dass eine Rechnung erst dann vollständig ist, wenn auch der dazugehörige Kunde eingetragen wurde.

Trigger werden auf dem Rechner ausgeführt, der die Datenbank-Engine enthält. Auf 4D Server werden Trigger innerhalb des aktuell laufenden Prozesses auf dem Server-Rechner ausgeführt, und nicht auf dem Client-Rechner. Mit 4D Server wird ein Trigger innerhalb des Benutzerprozesses ausgeführt, der die Operation in der Datenbank hervorruft. Der Trigger kann dagegen nicht auf die Prozessvariablen des Benutzerprozesses zugreifen.
Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* im Kapitel Trigger.

Serverprozeduren

Serverprozeduren sind Projektmethoden, die Methoden in Prozessen ausführen, die auf dem Server-Rechner laufen. Weitere Informationen finden Sie im Abschnitt Serverprozeduren.

Datenbankmethoden

Vier Datenbankmethoden werden allein auf dem Server-Rechner ausgeführt:

- Datenbankmethode On Server Startup
- Datenbankmethode On Server Shutdown
- Datenbankmethode On Server Open Connection
- Datenbankmethode On Server Close Connection

Vier weitere Datenbankmethoden lassen sich sowohl auf dem Server Rechner als auch auf dem Client Rechner ausführen:

- Datenbankmethode On Web Authentication
- Datenbankmethode On Web Connection
- Datenbankmethode On Backup Startup
- Datenbankmethode On Backup Shutdown

Weitere Informationen dazu finden Sie in den folgenden Abschnitten und im Handbuch *4th Dimension Programmiersprache* im Kapitel Datenbankmethoden.

4D Server und Variablen

• 4D Server verwaltet eine Tabelle Interprozessvariablen. Diese Variablen liegen auf dem Server-Rechner. Bei einer kompilierten Datenbank ist die Tabelle Interprozessvariablen für Server und alle Client-Rechner gleich, jeder Rechner hat jedoch einen eigenen Variableninhalt.

• Jede Serverprozedur hat eine eigene Tabelle Prozessvariablen. Bei einer kompilierten Datenbank ist die Tabelle Prozessvariablen für Server und alle Client-Rechner gleich, jeder Rechner hat jedoch einen eigenen Variableninhalt.

• In interpretierter Form können Datenbankmethoden und Trigger während jeder Ausführungsphase Prozessvariablen dynamisch erstellen und verwenden. Das geht nicht in kompilierter Form. Bei einer kompilierten Datenbank teilen sich Datenbankmethoden und Trigger eine gemeinsame Tabelle der Prozessvariablen. (Die Definition ist identisch mit der anderer Prozesse).

• Alle Serverprozeduren teilen sich die Interprozessvariablen auf dem Server (da sie dort ablaufen), unabhängig von welchem Client sie gestartet wurden. Trigger teilen sich die gleichen Interprozessvariablen wie die Serverprozeduren. Darüberhinaus greifen alle Trigger auf die gleichen Prozessvariablen zu, während jede Serverprozedur bzw. jeder Server prozess eigene Prozessvariablen erhalten.

4D Server und Mengen

Interprozess- und Prozessmengen werden auf dem Server-Rechner, lokale Mengen auf den Client-Rechnern verwaltet. Weitere Informationen finden Sie im Abschnitt 4D Server und Mengen.

Sie können mit Interprozess-, Prozess- oder lokalen Mengen arbeiten. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* im Kapitel Mengen.

• **Prozessmengen**: Eine Prozessmenge gilt nur für den Prozess, der die Menge erzeugt hat. Sie wird bei Beginn der Methodenausführung aktiv und bleibt es, bis die Methode vollständig ausgeführt ist. Dann ist die Prozessmenge gelöscht. Prozessmengen benötigen keine spezielle Kennzeichnung im Namen. UserSet und LockedSet sind Mengen mit reservierten Namen.
• **Interprozessmengen**: Eine Interprozessmenge gilt für alle Prozess auf der Arbeitsstation, welche diese Menge erzeugt hat. Ihr Name beginnt mit den Zeichen <> (“kleiner als” gefolgt von “größer als”).
Hinweis: Diese Schreibweise gilt für Windows und Macintosh. Auf Macintosh kann auch das Zeichen ◊ benutzt werden. Tippen Sie dazu bei gedrückter Umschalt- und Wahltaste den Buchstaben v ein.
• **Lokale Mengen/Client Mengen**: Der Name von lokalen bzw. Client Mengen beginnt mit dem Dollarzeichen ($). Auch die Systemmenge UserSet gehört zu den lokalen bzw. Client Mengen, obwohl Sie nicht mit einem Dollarzeichen $ beginnt.

Interprozess- und Prozessmengen werden auf dem Server Rechner verwaltet, lokale Mengen auf den Client-Rechnern.

Hinweis: Sie verwenden normalerweise Interprozess- und Prozessmengen, da diese die Verwaltung in der Client/Server Architektur optimieren.

4D Server und Befehle für Mengen

Dieser Abschnitt beschreibt das Verhalten von Befehlen für Mengen in der Client/Server Architektur bei der Ausführung auf dem Client Rechner:

• CREATE EMPTY SET
Eine leere Menge wird auf dem Server Rechner erstellt. Eine lokale Menge wird dann über das Netzwerk vom Server-Rechner auf den Client-Rechner kopiert. Interprozess- oder Prozessmengen bleiben auf dem Server-Rechner und werden auch dort gepflegt.

• CREATE SET
• CREATE SET FROM ARRAY

Eine Menge wird auf dem Server Rechner erstellt. Eine lokale Menge wird dann über das Netzwerk vom Server-Rechner auf den Client-Rechner kopiert. Interprozess- oder Prozessmengen bleiben auf dem Server-Rechner und werden auch dort gepflegt.

• USE SET
Eine lokale Menge wird zuerst vom Client-Rechner über das Netzwerk auf den Server-Rechner kopiert, und dann auf dem Server verwendet, um die Tabellenauswahl zu ändern. Interprozess- oder Prozessmengen werden lokal auf dem Server-Rechner verwendet, um die Tabellenauswahl zu ändern.

• SAVE SET
Eine lokale Menge wird lokal auf dem Client-Rechner gesichert.
Interprozess- oder Prozessmengen werden zuerst über das Netzwerk vom Server-Rechner auf den Client-Rechner kopiert und dann lokal auf dem Client-Rechner gesichert.

• LOAD SET
Eine lokale Menge wird von der Festplatte lokal auf den Client-Rechner geladen.
Interprozess- oder Prozessmengen werden zuerst von der Festplatte lokal auf den Client-Rechner geladen und dann über das Netzwerk vom Client-Rechner auf den Server-Rechner kopiert.

• CLEAR SET
• ADD TO SET
• REMOVE FROM SET
• Is in set
• Records in set

Diese fünf Befehle greifen auf lokale Mengen zu, die lokal auf dem Client-Rechner liegen. Bei Interprozess- oder Prozessmengen wird über das Netzwerk eine Anfrage an den Server-Rechner gesendet, um die Information zu erhalten bzw. die Aktion auszuführen.

• DIFFERENCE
• INTERSECTION
• UNION

Für diese drei Befehle müssen die dazugehörigen Mengenparameter auf demselben Rechner liegen. Folglich müssen entweder alle oder keiner davon lokale Mengen sein.

• COPY SET
Mit COPY SET können Sie jede Menge in eine andere kopieren. Sie können zum Beispiel eine lokale Menge in eine Interprozess- oder Prozessmenge kopieren. Die Menge wird dann über das Netzwerk vom Client-Rechner auf den Server-Rechner kopiert.

4D Server und die Menge UserSet

4D Client erstellt die Menge UserSet als lokale Menge (auch wenn der Name nicht mit einem $-Zeichen beginnt), um das Erstellen der Menge zu optimieren. UserSet enthält die Datensätze, die der Anwender in einer mit den Befehlen MODIFY SELECTION oder DISPLAY SELECTION erzeugten Liste angeklickt hat.

Wollen Sie UserSet als Parameter für die Befehle DIFFERENCE, INTERSECTION oder UNION übergeben, wenn die anderen Parameter Interprozess oder Prozessmengen sind, müssen Sie UserSet (eine lokale Menge) zuerst in eine Interprozess- oder Prozessmenge kopieren, um diese Menge mit dem Befehl benutzen zu können.

Beispiel:

```
ALL RECORDS ([Tabelle])
   ` Laß den Benutzer einige Datensätze auswählen
MODIFY SELECTION ([Tabelle];*)
   ` Prüfe, ob der Benutzer einige Datensätze ausgewählt hat
If (Records in set("UserSet")>0)
      ` Suche die Datensätze, die auszuschließen sind
   QUERY([Tabelle];[Tabelle]Flag#0)
      ` Erstelle eine Menge aus der sich ergebenden Auswahl
   CREATE SET([Tabelle];"auszuschließen")
   If (Application type = 4D Client)
         ` UserSet ist lokale Menge, kopiere sie in eine nicht-lokale Menge
      COPY SET ("UserSet";"Benutzerauswahl") ` ← über das Netzwerk kopiert
         ` Rufe DIFFERENCE auf und übergebe 3 nicht-lokale Parameter
                                                vom Typ Menge
      DIFFERENCE ("Benutzerauswahl";"auszuschließen";"Benutzerauswahl")
   Else
         ` Rufe DIFFERENCE
      DIFFERENCE ("UserSet";"auszuschließen";"Benutzerauswahl")
   End if
   CLEAR SET("auszuschließen")
   USE SET("Benutzerauswahl")
   CLEAR SET("Benutzerauswahl")
End if
```

Sie können die Operation auch mit folgendem Code ausführen:

```
ALL RECORDS ([Tabelle])
   ` Laß den Benutzer einige Datensätze auswählen
MODIFY SELECTION ([Tabelle];*)
   ` Prüfe, ob der Benutzer einige Datensätze ausgewählt hat
If (Records in set("UserSet")>0)
      ` Suche die Datensätze, die auszuschließen sind
   QUERY([Tabelle];[Tabelle]Flag#0)
      ` Erstelle eine Menge aus der sich ergebenden Auswahl
   CREATE SET([Tabelle];"auszuschließen")
   If (Application type = 4D Client)
         ` UserSet ist lokale Menge, kopiere sie in eine nicht-lokale Menge
      CREATE SET([Tabelle];"$auszuschließen") ` ← kopiere vom Server auf den Client
         ` Rufe DIFFERENCE auf und übergebe 3 nicht-lokale Parameter
                                                vom Typ Menge
      DIFFERENCE ("UserSet";"$Auszuschließen";"UserSet")
   Else
         ` Erstelle eine nicht lokale Menge aus der sich ergebenden Auswahl
      CREATE SET([Tabelle];"auszuschließen")
         ` Rufe DIFFERENCE
      DIFFERENCE ("UserSet";"auszuschließen";"Benutzerauswahl")
   End if
   CLEAR SET("$auszuschließen")
   USE SET("UserSet") ` ← kopiere vom Client zum Server
End if
```

Im ersten Beispiel werden drei Mengen erstellt und eine über das Netzwerk kopiert. Im zweiten Beispiel werden zwei Mengen erstellt und zwei über das Netzwerk kopiert. Wählen Sie je nach Ihrem Bedarf eine ähnliche Programmierung.

4D Server und die Menge LockedSet

LockedSet ist eine Menge, die auf dem Server-Rechner erstellt und gepflegt wird.

Referenz

COPY SET, Einführung, Einführung in Mengen.

Datenbankmethode On Server Startup

4D Programmiersprache

Version 6.8 (Geändert)

Die Datenbankmethode On Server Startup wird einmal auf dem Server-Rechner aufgerufen, wenn Sie eine Datenbank mit 4D Server öffnen. Die Datenbankmethode On Server Startup wird ausschließlich in der 4D Server Umgebung aufgerufen.

Die Datenbankmethode On Server Startup ist der richtige Ort, um:

- Interprozessvariablen zu initialisieren, die Sie während der gesamten 4D Server Sitzung verwenden.
- Serverprozeduren automatisch beim Öffnen der Datenbank zu starten.
- Voreinstellungen bzw. Einstellungen aus der letzten 4D Server Sitzung zu laden.
- Andere Aktionen auszuführen, die bei jedem Öffnen der Datenbank automatisch ablaufen sollen.

Um Code automatisch auf einem Client-Rechner auszuführen, wenn sich ein 4D Client am Server anmeldet, verwenden Sie die Datenbankmethode On Startup.

Referenz

Datenbankmethode On Server Shutdown, Datenbankmethoden, Import mit Serverprozeduren (Beispiel).

Datenbankmethode On Server Shutdown 4D Programmiersprache

Version 6.8 (Geändert)

Die Datenbankmethode On Server Shutdown wird einmal auf dem Server-Rechner aufgerufen, wenn Sie 4D Server beenden und somit die Datenbank verlassen. Die Datenbankmethode On Server Shutdown wird ausschließlich in der 4D Server Umgebung aufgerufen.

Eine Server Datenbank wird nur beendet, wenn der Benutzer auf dem Server den Befehl **Beenden** wählt oder eine Serverprozedur den Aufruf des Befehls QUIT 4D auslöst.

Beim Verlassen der Datenbank führt 4D folgende Aktionen durch:

• Ist keine Datenbankmethode On Server Shutdown vorhanden, beendet 4D Server ohne Unterscheidung der Reihe nach jeden Prozess.

• Ist eine Datenbankmethode On Server Shutdown vorhanden, führt 4D Server diese Methode in einem neu angelegten lokalen Prozess durch. Sie können so via Interprozesskommunikation anderen Prozesse mitteilen, die Ausführung zu beenden. Beachten Sie, dass 4D Server evtl. beendet wird – die Datenbankmethode On Server Shutdown kann zwar alle gewünschten Operationen zum Aufräumen oder Schließen durchführen, aber nicht das Beenden ansich verweigern, und so an einem gewissen Punkt enden.

Die Datenbankmethode On Server Shutdown ist der richtige Ort, um:
• Serverprozeduren zu beenden, die beim Öffnen der Datenbank automatisch starten.
• Voreinstellungen bzw. Einstellungen (lokal oder auf Festplatte) zu speichern, um sie beim Starten der nächsten Sitzung in der Datenbankmethode On Server Startup wiederzuverwenden.
• Andere Aktionen auszuführen,
• Andere Aktionen auszuführen, die bei jedem Beenden der Datenbank automatisch ablaufen sollen.

Wichtig: Verwenden Sie zum Schliessen von Serverprozeduren die Datenbankmethode On Server Shutdown, beachten Sie, dass der Server mit Ende der Ausführung der Datenbankmethode On Server Shutdown und nicht der Serverprozeduren abschaltet. Noch laufende Serverprozeduren werden abgebrochen.
Wollen Sie sicherstellen, dass die Serverprozeduren beenden, bevor der Server abschaltet, muss die Datenbankmethode On Server Shutdown eine Anweisung zum Beenden geben, z.B. über Testen einer Interprozessvariablen und die Zeit zum Beenden gewähren. Dazu können Sie eine Schleife von n-Sekunden oder einen Test durch eine andere Interprozessvariable einrichten.
Um Code automatisch auf einem Client-Rechner auszuführen, wenn sich ein 4D Client am Server abmeldet, verwenden Sie die Datenbankmethode On Exit.

Referenz

Datenbankmethode On Server Startup, Datenbankmethoden, Methoden.

Datenbankmethode On Server Open Connection 4D Programmiersprache

Version 6.8 (Geändert)

Wann wird die Datenbankmethode On Server Open Connection aufgerufen?

Die Datenbankmethode On Server Open Connection wird einmal auf dem Server-Rechner aufgerufen, wenn sich ein Client-Rechner anmeldet. Die Datenbankmethode On Server Open Connection wird ausschließlich in der 4D Server Umgebung aufgerufen.

4D Client

Mit 4D Client wird die Datenbankmethode On Server Open Connection immer aufgerufen, wenn:

- 4D Client sich anmeldet (weil der Prozess Benutzerumgebung startet)
- 4D Client die Designumgebung öffnet (weil der Prozess Design startet)
- 4D Client über den Befehl New Process einen nicht-lokalen Prozess startet
- Ein nicht-lokaler Prozess über ein Menü oder das Dialogfenster Methode ausführen aufgerufen wird.

Bei den Aktionen mit 4D Client starten zwei Prozesse —Einer auf dem Client-Rechner, der andere auf dem Server-Rechner. Auf dem Client-Rechner führt der Prozesscode aus und sendet Anfragen an 4D Server. Auf dem Server-Rechner verwaltet der Prozess die Datenbankumgebung für den Client-Prozess (z.B. akutelle Auswahlen für den Benutzerprozess) und beantwortet Anfragen, die von dem Prozess auf dem Client-Rechner gesendet wurden. Von daher wird der Prozess, der auf dem Server läuft, **Verbindungsprozeß** genannt. Er führt im Grunde genommen dasselbe aus, wie der Prozess auf dem Rechner im Einzelplatzbetrieb, mit dem Unterschied, dass er über das Netzwerk läuft.

4D Insider

Melden Sie sich von 4D Insider aus an 4D Server an, startet auf dem Server-Rechner ein Verbindungsprozess für die Arbeitsumgebung von 4D Insider. Dieser Prozess beantwortet die Anfragen, die 4D Insider sendet.

Auf 4D Open basierende Anwendungen

Jedes Mal, wenn sich eine auf 4D Open-basierende Anwendung an 4D Server anmeldet, startet ein Verbindungsprozess auf dem Server. Dieser Prozess beantwortet die Anfragen, die über 4D Open gesendet werden und verwaltet den Datenbankbereich dieser Verbindung (z.B. aktuelle Auswahl).

Wichtig: Web Verbindungen lösen nicht die Datenbankmethode On Server Open Connection aus. Meldet sich ein Web Browser an 4D Server an, startet die Datenbankmethode On Web Autentication -sofern vorhanden und/oder die Datenbankmethode On Web Connection. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* im Kapitel *Web Server*.

Wichtig: Startet eine Serverprozedur, wird nicht die Datenbankmethode On Server Open Connection ausgelöst. Serverprozeduren sind Serverprozesse und keine Verbindungsprozesse. Sie führen Code auf dem Server-Rechner aus, beantworten jedoch keine Anfragen von 4D Clients oder anderen Clients an 4D Server.

Wie wird die Datenbankmethode On Server Open Connection aufgerufen?

Die Datenbankmethode On Server Open Connection wird auf dem 4D Server-Rechner innerhalb des Verbindungsprozesses ausgeführt, der die Anweisung gibt, die Methode aufzurufen.

Beispiel: Meldet sich ein 4D Client an eine Datenbank auf 4D Server an, starten die Prozesse Benutzerumgebung und Design und Clienteintragung für diesen Client. Die Datenbankmethode On Server Open Connection wird also dreimal ausgeführt —einmal innerhalb des Verbindungsprozesses Benutzerumgebung, einmal innerhalb des Verbindungsprozesses Clienteintragung und einmal innerhalb des Verbindungsprozesses Design. Sind diese Prozesse jeweils der sechste, siebte und achte gestartete Prozeß auf dem Server-Rechner, gibt die Funktion Current process jeweils 6, 7 und 8 zurück, wenn Sie Current process von der Datenbankmethode On Server Open Connection aufrufen.

Beachten Sie, dass die Datenbankmethode On Server Open Connection auf dem Server-Rechner ausgeführt wird. Sie läuft im Verbindungsprozess ab, der auf dem Server läuft, unabhängig vom Prozess auf der Client-Seite. Außerdem ist der Verbindungsprozeß in dem Moment, wo die Methode ausgelöst wird, noch nicht benannt. (Der Befehl PROCESS PROPERTIES gibt zu diesem Zeitpunkt nicht den Namen des Verbindungsprozesses zurück).

Die Datenbankmethode On Server Open Connection hat keinen Zugriff auf die Tabelle Prozessvariablen des Prozesses auf der Client-Seite, da diese Tabelle auf dem Client-Rechner und nicht auf dem Server-Rechner liegt.

Greift die Datenbankmethode On Server Open Connection in einer interpretierten Datenbank auf eine Prozessvariable zu, läuft sie in einer eigenen und dynamisch erstellten Tabelle Prozessvariablen für den Verbindungsprozess ab. Da die On Server Close Connection Database Method unter Umständen im gleichen Verbindungsprozeß ausgelöst wird, nehmen Sie vielleicht an, dass Sie Informationen zwischen den beiden Methoden über Prozessvariablen verwalten können. **Das funktioniert nicht im kompilierten Modus**.

In einer komplierten Datenbank teilt sich die Datenbankmethode On Server Open Connection zusammen mit anderen Datenbankmethoden sowie Triggern eine gemeinsame Tabelle Prozessvariablen, die auf dem Server-Rechner verwaltet wird. Auf diese Weise kann der kompilierte Code laufen und der Speicherverbrauch wird reduziert. Sie können einerseits von einer Datenbankmethode oder einem Trigger aus auf jede Prozessvariable zugreifen. Andererseits wird Speicherplatz gespart, denn eine eigene Prozesstabelle pro Datenbankmethode sowie Trigger würde sowohl Speicherplatz als auch Startzeit beanspruchen.
Fazit: Stützen Sie sich beim Ausführen der Datenbankmethode On Server Open Connection und On Server Close Connection Database Method nicht auf Prozessvariablen. Verwenden Sie Daten, die in Interprozeßvariablen oder in einer Tabelle gespeichert sind.

4D Server übergibt der Datenbankmethode On Server Open Connection drei Parameter vom Typ Lange Ganzzahl und erwartet ein Ergebnis vom Typ Lange Ganzzahl. Die Methode muss deshalb ausdrücklich mit drei Parametern vom Typ Lange Ganzzahl deklariert werden und dem Funktionsergebnis vom Typ Lange Ganzzahl:

```
C_LONGINT($0;$1;$2;$3)
```

Geben Sie keinen Wert in $0 zurück, d.h. die Variable ist undefiniert oder gegen Null initialisiert, nimmt 4D Server an, daß die Datenbankmethode die Verbindung akzeptiert. Akzeptieren Sie die Verbindung nicht, geben Sie in $0 einen Wert zurück, der nicht Null ist.

Die drei Parameter für die Datenbankmethode geben folgende Informationen zurück:

Parameter	Beschreibung
$1	ID Nummer für Benutzer, die 4D Server intern zur Identifikation des Benutzers verwendet.
$2	ID Nummer für Verbindung, die 4D Server intern zur Identifikation einer Verbindung verwendet.
$3	ID Nummer für Netzwerkprotokoll, die 4D Server intern verwendet.

Hinweis: Ab 4D Server Version 6.8 gibt der Parameter $3 immer 2 oder 29 zurück.

Diese drei ID Nummern sind nicht direkt als Informationsquellen einsetzbar, z.B. um sie in einem 4D Befehl als Parameter zu übergeben. Sie ermöglichen jedoch, einen Verbindungsprozeß zwischen der Datenbankmethode On Server Open Connection und der Datenbankmethode On Server Close Connection eindeutig zu identifizieren. Die Kombination dieser drei Werte ist zu jedem Moment einer Server Sitzung einmalig. Ist diese Information in einer Interprozeß-Array bzw. Tabelle gespeichert, können beide Datenbankmethoden Informationen austauschen.

Im Beispiel am Ende dieses Abschnitts, verwenden die beiden Datenbankmethoden diese Information, um Datum und Uhrzeit für Beginn und Ende einer Verbindung zum gleichen Datensatz einer Tabelle zu speichern.

Beispiele

1. Dieses Beispiel zeigt, wie mit der Datenbankmethode On Server Open Connection und der On Server Close Connection Database Method in der Datenbank ein Logbuch der Verbindung verwaltet wird. Die Tabelle [Server Log] wird zum Auffinden der Verbindungsprozesse verwendet.

Logbuch	
Log Nr	**L**
Anmeldedatum	**D**
Anmeldezeit	**U**
Abmeldedatum	**D**
Abmeldezeit	**U**
Benutzer Nr	**L**
Verbindung Nr	**L**
NC Nr	**L**
Prozeß Nr	**L**
Prozeß Name	**A**

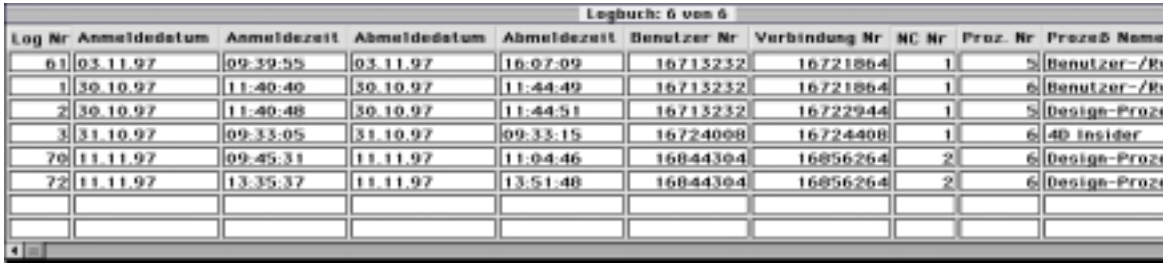
Die hier gespeicherte Information wird von der Datenbankmethode On Server Open Connection und der Datenbankmethode On Server Close Connection wie folgt verwaltet:

```
    ` Datenbankmethode On Server Open Connection
C_LONGINT($0;$1;$2;$3)
    ` Erstelle eine Datensatz [Logbuch]
CREATE RECORD([Logbuch])
[Logbuch]Log Nr:=Sequence number([Logbuch])
    ` Sichere Anmeldedatum und Anmeldezeit
[Logbuch]Anmeldedatum:=Current date
[Logbuch]Anmeldezeit:=Current time
    ` Sichere die Verbindungsinformation
[Logbuch]Benutzer Nr:=$1
[Logbuch]Verbindung Nr:=$2
[Logbuch]NC Nr:=$3
SAVE RECORD([Logbuch])
```

```
    ` Gibt keinen Fehler zurück, so dass die Verbindung weiterlaufen kann
$0:=0

    ` Datenbankmethode On Server Close Connection
C_LONGINT($1;$2;$3)
    ` Finde wieder Datensatz [Logbuch]
QUERY([Logbuch];[Logbuch]Benutzer Nr=$1;*)
QUERY([Logbuch]; & ;[Logbuch]Verbindung Nr=$2;*)
QUERY([Logbuch]; & ;[Logbuch]NC Nr=$3;*)
QUERY([Logbuch]; & ;[Logbuch]Prozeß Nr=0)
    ` Sichere Abmeldedatum und -zeit
[Logbuch]Abmeldedatum:=Current date
[Logbuch]Abmeldezeit:=Current time
    ` Sichere die Prozessinformation
[Logbuch]Prozeß Nr:=Current process
PROCESS PROPERTIES([Logbuch]Prozeß Nr;$vsProzName;$vlProzStatus;$vlProzZeit)
[Logbuch]Prozeß Name:=$vsProzName
SAVE RECORD([Logbuch])
```

Nachfolgend sehen Sie einige Einträge in [Logbuch] mit mehreren 4D Client Anbindungen und einer 4D Insider Anbindung:



Logbuch: 6 von 6

Log Nr	Anmeldedatum	Anmeldezeit	Abmeldedatum	Abmeldezeit	Benutzer Nr	Verbindung Nr	NC Nr	Proz. Nr	Prozeß Name
61	03.11.97	09:39:55	03.11.97	16:07:09	16713232	16721864	1	5	Benutzer-/R
1	30.10.97	11:40:40	30.10.97	11:44:49	16713232	16721864	1	6	Benutzer-/R
2	30.10.97	11:40:48	30.10.97	11:44:51	16713232	16722944	1	5	Design-Proz
3	31.10.97	09:33:05	31.10.97	09:33:15	16724008	16724408	1	6	4D Insider
70	11.11.97	09:45:31	11.11.97	11:04:46	16844304	16856264	2	6	Design-Proz
72	11.11.97	13:35:37	11.11.97	13:51:48	16844304	16856264	2	6	Design-Proz

2. Im folgenden Beispiel kann zwischen 2 und 4 Uhr A.M. keine neue Verbindung hergestellt werden:

```
` Datenbankmethode On Server Open Connection
C_LONGINT($0;$1;$2;$3)

If((?02:00:00?<=Current time)&(Current time<?04:00:00?))
   $0:=22000
Else
   $0:=0
End if
```

Referenz

Datenbankmethode On Server Close Connection, Datenbankmethoden.

Datenbankmethode On Server Close Connection 4D Programmiersprache

Version 6.8 (Geändert)

Die Datenbankmethode On Server Close Connection wird jedes Mal auf dem Server-Rechner aufgerufen, wenn ein Verbindungsprozess endet.

Analog zur Datenbankmethode On Server Open Connection übergibt 4D Server in der Datenbankmethode On Server Close Connection drei Parameter vom Typ Lange Ganzzahl, er erwartet jedoch kein Ergebnis.

Die Methode muss deshalb ausdrücklich mit drei Parametern vom Typ Lange Ganzzahl definiert werden:

C_LONGINT($1;$2;$3)

Die drei Parameter für die Datenbankmethode geben folgende Informationen zurück:

Parameter	**Beschreibung**
$1	ID Nummer für Benutzer, die 4D Server intern zur Identifikation des Benutzers verwendet.
$2	ID Nummer für Verbindung, die 4D Server intern zur Identifikation einer Verbindung verwendet.
$3	ID Nummer für Netzwerkkomponente, die 4D Server intern verwendet.

Hinweis: Ab 4D Server Version 6.8, gibt der Parameter $3 immer 2 oder 29 zurück.

Die Datenbankmethode On Server Close Connection ist das exakte Gegenstück zur Datenbankmethode On Server Open Connection. Weitere Informationen und eine Beschreibung der **Verbindungsprozesse**, finden Sie im vorigen Abschnitt.

Beispiel

Siehe erstes Beispiel für die Datenbankmethode On Server Open Connection.

Referenz

Datenbankmethode On Server Open Connection, Datenbankmethoden.

8

Serverprozeduren

Was ist eine SQL Serverprozedur?

Der Ausdruck Serverprozedur stammt aus der Server-Welt, die auf SQL basiert. Sendet eine Arbeitsstation eine Anfrage an einen SQL Server, sendet sie die Anfrage als Volltext in SQL Sprache an diesen Server. Die Anfrage wird vor der Ausführung auf dem SQL Server analysiert und interpretiert. Ist der Quellcode umfangreich und wird die Anfrage während einer Sitzung mehrere Male gesendet, beansprucht das natürlich viel Zeit für die Übertragung über das Netzwerk und die stets neue Analyse und Interpretation.

Deshalb sollte ein Weg gefunden werden, diese Anfrage zuerst über das Netzwerk zu senden, zu analysieren und zu interpretieren und nur noch auszuführen, wenn sie von der Arbeitsstation gesendet wird. Die Lösung war, den Quellcode der Anfrage (mit anderen Worten eine Prozedur) auf der Server-Seite zu lassen; die Arbeitsstation sendet dann als Anfrage nur den Namen dieser Prozedur. Da diese Prozedur auf dem Server gespeichert wird, nennen wir solche Prozeduren "Serverprozeduren".

Beachten Sie, dass eine Serverprozedur, die auf SQL basiert, Parameter von einer Arbeitsstation empfangen kann, die damit verbundenen Tasks synchron oder asynchron ausführt und evtl. ein Ergebnis an die Arbeitsstation zurückgibt. Löst eine Arbeitsstation die Ausführung einer Serverprozedur aus, überlässt die Station die Ausführung von Code bis zu einem gewissen Maß dem Server-Rechner.

Was ist eine 4D Server Serverprozedur?

Selbst wenn wir den gängigen Namen benutzen, bieten Serverprozeduren unter 4D Server weit mehr als reguläre Serverprozeduren.

Mit einer 4D Funktion, wie z.B. New process können Sie einen Benutzerprozess starten, in dem eine Methode abläuft. Diese Methode heißt **Prozessmethode**. Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* im Abschnitt Projektmethoden.

Dasselbe können Sie mit 4D Server auf einer Arbeitsstation durchführen. Sie können zusätzlich mit der Funktion Execute on server auf dem Server-Rechner einen Benutzerprozess starten, in dem eine Methode abläuft. Diese Methode heißt **Serverprozedur**. Der Prozess, der auf dem Server-Rechner gestartet wird, heißt ebenfalls **Serverprozedur**.

Wichtig: Der Hauptunterschied zwischen einer Serverprozedur, die auf SQL-basiert und einer Serverprozedur unter 4D Server ist, dass im ersten Fall eine SQL Prozedur, im zweiten Fall ein 4D Prozess abläuft.

Architektur der 4D Serverprozeduren

Eine Serverprozedur hat wie ein normaler Prozess eine eigene Umgebung:

• **Aktuelle Auswahl pro Tabelle**: Jede Serverprozedur hat eine eigene aktuelle Auswahl. Eine Tabelle kann in verschiedenen Serverprozeduren auch verschiedene Auswahlen haben.
• **Aktueller Datensatz pro Tabelle**: Jede Tabelle kann in jeder Serverprozedur einen anderen aktuellen Datensatz haben.
• **Variablen**: Jede Serverprozedur hat ihre eigenen Prozessvariablen. Prozessvariablen werden nur im Bereich der jeweiligen Serverprozedur erkannt.
• **Standardtabelle**: Jede Serverprozedur hat eine eigene Standardtabelle.
• **Prozeßmengen**: Jede Serverprozedur hat ihre eigenen Prozessmengen.
• **On Error Call**: Jede Serverprozedur hat ihre eigene Methode **On Error Call**.
• **Debugger Fenster**: Jede Serverprozedur kann ein eigenes Debugger Fenster haben.

Eine Serverprozedur kann in der Benutzeroberfläche Fenster öffnen und Daten anzeigen (z.B. DISPLAY RECORD). Sie kann jedoch im Gegensatz zu Benutzerprozessen auf der Arbeitsstation keine Dateneingabe auslösen (z.B. ADD RECORD), da es keinen Eingabe-Kernel auf dem Server-Rechner gibt.

Sie können soviele Serverprozeduren starten wie es Hardware und Speicher des Systems zulassen. Im Grunde genommen antwortet der 4D Server-Rechner nicht nur auf Anfragen der Arbeitsstation und von Web Browsern, er führt auch Prozesse aus, die mit anderen Prozessen interagieren, die auf dem Server-Rechner und Client-Rechnern laufen.

Ebenso wie 4th Dimension und 4D Client eine Multi-Tasking Umgebung für Benutzerprozesse auf einer Arbeitsstation bieten, bietet 4D Server eine Multi-Tasking Umgebung für Serverprozeduren. 4D Server verwaltet zum Beispiel eine Tabelle mit Interprozessvariablen, die Serverprozeduren für die Interprozesskommunikation benutzen können.

Was macht eine Serverprozedur?

Fast alles, was im Handbuch *4th Dimension Programmiersprache* über die Funktionsweise von Prozessen und Befehlen beschrieben wird, gilt auch für Serverprozeduren. Davon ausgenommen sind die Eingabe von Daten und einige Druckoptionen.

Eine Serverprozedur kann Datensätze hinzufügen, suchen, sortieren, aktualisieren und löschen. Sie kann Mengen und Auswahlen verwenden, auf Dokumente auf der Festplatte zugreifen, mit BLOBs u.v.m. arbeiten. Der einzige Unterschied besteht darin, dass etwas nicht auf dem 4D Client-Rechner, sondern auf dem Server-Rechner ausgeführt wird.

Ein großer Vorteil ist, dass eine Serverprozedur lokal auf dem Server-Rechner ausgeführt wird, also dort, wo auch die Datenbank-Engine liegt. Der Befehl APPLY TO SELECTION ist innerhalb einer Serverprozedur effizienter als über das Netzwerk. Das Beispiel im Abschnitt Import mit Serverprozeduren (Beispiel) zeigt, wie stark sich die Leistung über clever eingesetzte Serverprozeduren optimieren lässt.

Der wichtigste Vorteil von Serverprozeduren sind jedoch die neuen Möglichkeiten, die sich für 4D Server ergeben. Mit Serverprozeduren können Sie Ihre eigenen 4D Server Dienste einrichten. Ihrer Phantasie sind keine Grenzen gesetzt. Das Beispiel im Abschnitt Service mit Serverprozeduren (Beispiel) zeigt eine Serverprozedur, die Clients mit Information über 4D Server bzw. den Server-Rechner ausstattet. Sie können z.B. die Laufwerke des Server-Rechners auflisten. Dieses Beispiel lässt sich leicht erweitern, so dass ein Client auch Information über ein Verzeichnis oder ein Dokument erhält.

Auf dem Server nicht zulässige Befehle

Enthält eine Serverprozedur einen dieser Befehle, erscheint eine Meldung, dass der Befehl nicht auf 4D Server ausgeführt werden kann. Der Fehler #67 wird zurückgegeben. Sie können ihn mit dem Befehl ON ERR CALL verwalten.

ACCUMULATE
ADD RECORD
ADD SUBRECORD
APPEND MENU ITEM
BREAK LEVEL
CALL PROCESS
CHANGE LICENSES
Count menu items
Count menus
CREATE DATA FILE
DELETE MENU ITEM
DISABLE MENU ITEM
DISPLAY SELECTION
EDIT ACCESS
ENABLE MENU ITEM
FILTER EVENT
Get menu item
Get menu item key
Get menu item mark
Get menu item style

Get menu title
GRAPH TABLE
HIDE MENU BAR
INSERT MENU ITEM
Level
Menu selected
MODIFY RECORD
MODIFY SELECTION
MODIFY SUBRECORD
ON EVENT CALL
OPEN DATA FILE
Open external window
PAGE BREAK
PAGE SETUP
PRINT FORM
PRINT LABEL
PRINT SETTINGS
Printing page
QR REPORT
QUERY BY EXAMPLE
REMOVE PICTURE FROM LIBRARY
SET ABOUT
SET MENU ITEM
SET MENU ITEM KEY
SET MENU ITEM MARK
SET MENU ITEM STYLE
SET PICTURE TO LIBRARY
SET PRINT PREVIEW
SHOW MENU BAR
Subtotal

Auf dem Server ungeeignete Befehle

Enthält eine Serverprozedur einen dieser Befehle, kann der Server blockiert werden. Es kann ein Fehler erzeugt werden oder nicht das gewünschte Ergebnis eintreten. Sie erhalten keine spezifische Fehlermeldung. Vermeiden Sie deshalb diese Befehle in Serverprozeduren.

ACCEPT
Activated
ADD DATA SEGMENT
After
APPEND TO CLIPBOARD
APPEND TO LIST
Before

BLOB to list
BRING TO FRONT
BUTTON TEXT
CANCEL
CHANGE CURRENT USER
CHANGE PASSWORD
CLEAR CLIPBOARD
CLEAR LIST
Copy list
Count list items
Count screens
Create document (1)
Create resource file (1)
Current form page
Current form table
Current user
C_GRAPH
Deactivated
DELETE LIST ITEM
DELETE USER
DIALOG
DISABLE BUTTON
DRAG AND DROP PROPERTIES
DRAG WINDOW
Drop position
During
ENABLE BUTTON
ERASE WINDOW
EXPORT DATA (1)
FILTER KEYSTROKE
Find window
FIRST PAGE
Focus object
FONT
FONT LIST
Font name
Font number
FONT SIZE
FONT STYLE
Form event
Frontmost process
Frontmost window
GET CLIPBOARD
Get edited text

GET FORM PROPERTIES
GET GROUP LIST
GET GROUP PROPERTIES
GET HIGHLIGHT
GET LIST ITEM
GET LIST ITEM PROPERTIES
GET LIST PROPERTIES
GET MOUSE
GET OBJECT RECT
GET PICTURE FROM CLIPBOARD
Get text from clipboard
GET USER LIST
GET USER PROPERTIES
GET WINDOW RECT
Get window title
GOTO AREA
GOTO PAGE
GRAPH SETTINGS
HIDE PROCESS
HIDE TOOL BAR
HIDE WINDOW
HIGHLIGHT RECORDS
HIGHLIGHT TEXT
IMPORT DATA (1)
In break
In footer
In header
INPUT FORM
INSERT LIST ITEM
INVERT BACKGROUND
Is a list
Is user deleted
Keystroke
LAST PAGE
List item parent
List item position
LIST TO BLOB
Load list
MAXIMIZE WINDOW
Menu bar height
Menu bar screen
MINIMIZE WINDOW
Modified
MOVE OBJECT

New list
NEXT PAGE
Next window
Old
Open document (1)
Open resource file (1)
ORDER BY (2)
OUTPUT FORM
Outside call
PRINT RECORD (2)
PRINT SELECTION (2)
Pop up menu
POST CLICK
POST EVENT
POST KEY
PREVIOUS PAGE
QUERY BY FORMULA (2)
QUERY (2)
REDRAW
REDRAW LIST
REDRAW WINDOW
REGISTER CLIENT
REJECT
SAVE LIST
SCREEN COORDINATES
SCREEN DEPTH
Screen height
Screen width
Select folder
SELECT LIST ITEMS BY POSITION
SELECT LIST ITEMS BY REFERENCE
SELECT LOG FILE
Selected list items
Self
SET CHOICE LIST
SET COLOR
SET CURSOR
SET ENTERABLE
SET FIELD TITLES
SET FILTER
SET FORMAT
Set group properties
SET LIST ITEM
SET LIST ITEM PROPERTIES

SET LIST PROPERTIES
SET PICTURE TO CLIPBOARD
SET RGB COLORS
SET SCREEN DEPTH
SET TABLE TITLES
SET TEXT TO CLIPBOARD
SET TIMER
Set user properties
SET VISIBLE
SET WINDOW RECT
Shift down
SHOW PROCESS
SHOW WINDOW
SORT LIST
Test clipboard
User in group
Validate password
Window kind
WINDOW LIST
Window process

(1) Nur wenn der erste Parameter ein leerer String ist.
(2) Nur wenn die Syntax die Anzeige eines Dialogfensters auslöst, z.B. SORT ([Tabelle]).

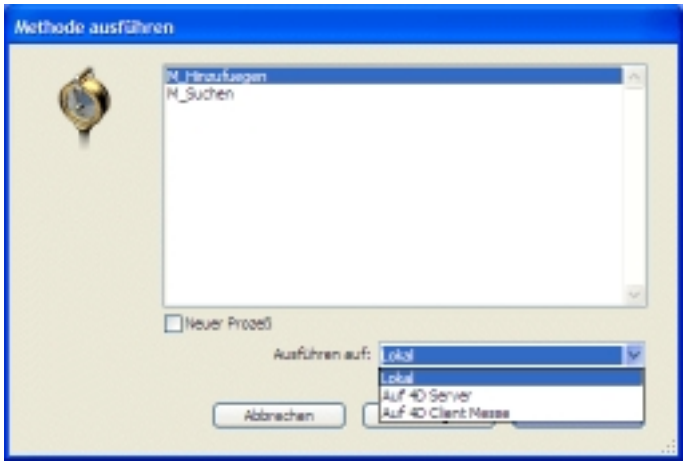
Befehle ohne Auswirkung auf dem Server

Diese Befehle haben keine Auswirkung in Serverprozeduren auf dem Server. Sie erhalten keine Fehlermeldung.

GRAPH
MENU BAR
MESSAGES OFF
MESSAGES ON
SHOW TOOL BAR

Eine Serverprozedur starten

• Mit 4D Client können Sie eine Serverprozedur im Dialogfenster Methode ausführen manuell starten:



Sie können sie auf 4D Server oder einem bzw. mehreren 4D Client Rechnern ausführen. Beachten Sie, dass nur die Clients in der Liste erscheinen, welche Sie zuvor registriert haben. Weitere Informationen dazu finden Sie im Abschnitt 4D Client registrieren und unter dem Befehl REGISTER CLIENT.

• Über Code in 4D Client: Rufen Sie die Funktion Execute on server auf.

• Über eine Methode, die auf 4D Server ausgeführt wird (Server Datenbankmethode oder Serverprozedur): Wählen Sie dazu die Funktion Execute on server oder New process oder EXECUTE ON CLIENT.

Interprozess-Kommunikation zwischen Serverprozeduren und Benutzerprozessen

Serverprozeduren können miteinander kommunizieren durch:

• Interprozessvariablen
• lokale oder globale Semaphoren
• Datensätze
• Interprozessmengen und temporäre Auswahlen auf Interprozessebene
• die Befehle GET PROCESS VARIABLE, SET PROCESS VARIABLE und VARIABLE TO VARIABLE

Weitere Informationen dazu finden Sie im Handbuch *4th Dimension Programmiersprache* in den entsprechenden Abschnitten. Nochmal zur Erinnerung: Denken Sie daran, dass die 4D Befehle im Bereich des Server-Rechners genauso arbeiten, wie im Bereich des Client-Rechners.

Hinweis: Die Befehle CALL PROCESS und Outside call haben auf dem Server-Rechner keine Bedeutung, da Serverprozeduren keine Benutzeroberfläche mit Dateneingabe haben.

Es gibt jedoch ein anderes wichtiges Feature: Benutzerprozesse, die auf dem Client-Rechner laufen, können mit den Befehlen GET PROCESS VARIABLE, SET PROCESS VARIABLE und VARIABLE TO VARIABLE Prozessvariablen (*) von einer Serverprozedur lesen und schreiben.

(*) ebenso wie die Interprozessvariable auf dem Server-Rechner.

Wichtig: Die Prozesskommunikation zwischen mehreren Rechnern über die Befehle GET PROCESS VARIABLE, SET PROCESS VARIABLE und VARIABLE TO VARIABLE ist nur vom Client zum Server möglich. Es ist immer ein Client-Prozess, der die Variablen einer Serverprozedur liest oder schreibt.

Referenz

Import mit Serverprozeduren (Beispiel), Service mit Serverprozeduren (Beispiel)

Folgendes Beispiel zeigt, wie sich der Datenimport im Client/Server-Betrieb enorm beschleu nigen lässt. Mit der Methode Reguläre Projektmethode Import können Sie testen, wie lange es dauert, mit dem Befehl IMPORT TEXT Datensätze auf der Arbeitsstation zu importieren:

```
   ` Reguläre Projektmethode Import
$vhDocRef:=Open document("")
If (OK=1)
   CLOSE DOCUMENT($vhDocRef)
   INPUT FORM([Table1];"Import")
   $vhStartTime:=Current time
   IMPORT TEXT([Table1];Document)
   $vhEndTime:=Current time
   ALERT("Es dauerte "+String(0+($vhEndTime-$vhStartTime))+" Sekunden.")
End if
```

Mit dieser Methode analysiert 4D Client die Textdatei, erstellt dann für jeden Datensatz einen neuen Datensatz, füllt die Datenfelder mit den importierten Daten und sendet den Datensatz zum Server-Rechner, damit er der Datenbank hinzugefügt wird. Demzufolge laufen viele Anfragen über das Netzwerk. Diese Operation läßt sich mit einer Server-prozedur optimieren, da sie dann lokal auf dem Server-Rechner ausgeführt wird. Der Client-Rechner lädt das Dokument in ein BLOB, startet dann eine Serverprozedur, die das BLOB als Parameter übergibt. Die Serverprozedur speichert das BLOB in einem Dokument auf der Festplatte des Server-Rechners und importiert dann das Dokument lokal. Der Datenimport wird demnach lokal ausgeführt (mit der Geschwindigkeit für Einzelplatz), da die meisten Anfragen über das Netzwerk eliminiert wurden.

Hier ist die Projektmethode CLIENT IMPORT. Sie wird auf dem Client-Rechner ausgeführt und ruft die u.a. Serverprozedur SERVER IMPORT auf:

```
   ` Projektmethode CLIENT IMPORT
   ` CLIENT IMPORT ( Pointer ; String )
   ` CLIENT IMPORT ( -> [Table] ; Input form )

C_POINTER($1)
C_STRING(31;$2)
C_TIME($vhDocRef)
C_BLOB($vxData)
C_LONGINT(spErrCode)
```

```
   ` Wähle das zu importierende Dokument
$vhDocRef:=Open document("")
If (OK=1)
      ` Wurde ein Dokument ausgewählt, schließe es
   CLOSE DOCUMENT($vhDocRef)
   $vhStartTime:=Current time
      ` Versuche, es in den Speicher zu laden
   DOCUMENT TO BLOB(Document;$vxData)
   If (OK=1)
         ` Konnte das Dokument in das BLOB geladen werden,
         ` Starte die Serverprozedur, die die Daten auf dem Server-Rechner importiert.
      $spProcessID:=Execute on server("SERVER IMPORT";32*1024;
                              "Server Import Services";Table($1);$2;$vxData)
         ` Wir benötigen das BLOB nicht länger in diesem Prozeß
      CLEAR VARIABLE($vxData)
         ` Warte bis die Operation durch die Serverprozedur vollständig ausgeführt ist
      Repeat
         DELAY PROCESS(Current process;300)
         GET PROCESS VARIABLE($spProcessID;spErrCode;spErrCode)
         If (Undefined(spErrCode))
            ` Hinweis: Hat die Serverprozedur nicht ihre eigene Instanz der Variable
            ` spErrCode initialisiert, wird evtl. eine undefinierte Variable zurückgegeben
            spErrCode:=1
         End if
      Until (spErrCode<=0)
         ` Teile der Serverprozedur mit, daß wir bestätigen
      spErrCode:=1
      SET PROCESS VARIABLE($spProcessID;spErrCode;spErrCode)
      $vhEndTime:=Current time
      ALERT("es dauerte "+String(0+($vhEndTime-$vhStartTime))+" Sekunden.")
   Else
      ALERT("Es gibt nicht genügend Speicher, um das Dokument zu laden.")
   End if
End if
```

Hier ist die Projektmethode SERVER IMPORT, ausgeführt als Serverprozedur:

```
    ` Projektmethode SERVER IMPORT
    ` SERVER IMPORT ( Long ; String ; BLOB )
    ` SERVER IMPORT ( Table Number ; Input form ; Import Data )

C_LONGINT($1)
C_STRING(31;$2)
C_BLOB($3)
C_LONGINT(spErrCode)

   ` Operation ist noch nicht beendet, setze spErrCode auf 1
spErrCode:=1
$vpTable:=Table($1)
INPUT FORM($vpTable->;$2)
$vsDocName:="Import File "+String(1+Random)
DELETE DOCUMENT($vsDocName)
BLOB TO DOCUMENT($vsDocName;$3)
IMPORT TEXT($vpTable->;$vsDocName)
DELETE DOCUMENT($vsDocName)
   ` Operation ist beendet, setze spErrCode auf 0
spErrCode:=0
   ` Warte, bis der anfragende Client das Ergebnis erhalten hat.
Repeat
   DELAY PROCESS(Current process;1)
Until (spErrCode>0)
```

Sind diese beiden Projektmethoden in einer Datenbank integriert, können Sie den Import über eine Serverprozedur ausführen, wenn Sie beispielsweise schreiben:

```
CLIENT IMPORT (->[Table1];"Import")
```

Bei einigen Benchmarks werden Sie feststellen, daß der Datensatzimport mit dieser Methode bis zu 60 mal schneller läuft als über den regulären Import.

Referenz

Execute on server, GET PROCESS VARIABLE, Serverprozeduren, Service mit Serverprozeduren (Beispiel), SET PROCESS VARIABLE.

Service mit Serverprozeduren (Beispiel)

Version 6.8 (Geändert)

Im Beispiel aus dem vorigen Abschnitt Import mit Serverprozeduren (Beispiel) startet oder endet eine Serverprozedur immer, wenn ein Datenimport angefordert wird. In diesem Beispiel startet eine Serverprozedur automatisch beim Starten der Serverdatenbank und kann von jedem Client, der sich an die Datenbank anmeldet, beliebig beendet oder wieder gestartet werden. Sobald die Serverprozedur läuft, kann sie asynchron auf die verschiedenen Anfragen antworten, welche die an die Datenbank angeschlossenen Clients senden.

Der vorige Abschnitt zeigt, wie Sie einen in 4D Server bestehenden Service zur Optimierung von Operationen integrieren können. Dieser Abschnitt erläutert, wie Sie neue Services integrieren können, die für alle 4D Clients verfügbar sind. Sie können das Beispiel auch als Vorlage zum Einrichten eigener Services verwenden.

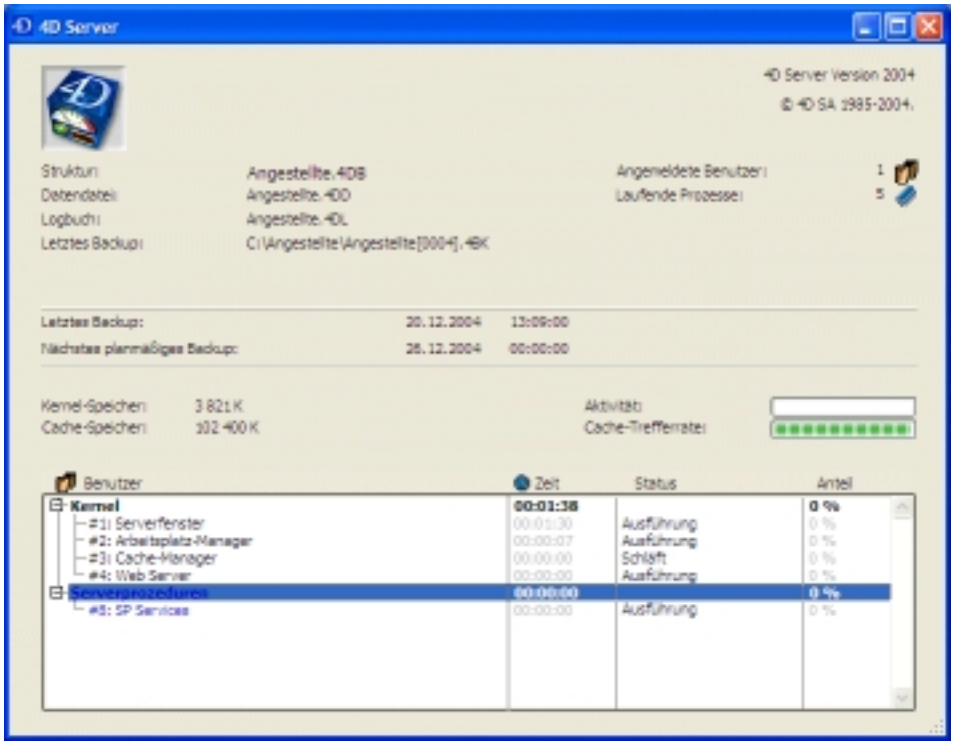
Automatisches Starten der Serverprozedur

Die Serverprozedur wird automatisch gestartet mit der Datenbankmethode On Server Startup:

```
` Datenbankmethode On Server Startup
START SP SERVICES
```

Da die Datenbankmethode On Server Startup die Projektmethode SP SERVICES als Serverprozedur startet, läuft SP SERVICES, sobald die Datenbank mit 4D Server gestartet wird, unabhängig davon, ob derzeit Clients an die Server-Datenbank angemeldet sind.

Das hier abgebildete Prozessfenster von 4D Server zeigt die aktive Serverprozedur, ohne dass ein Client angemeldet ist.



Hinweis: Alle Serverprozeduren gelten als ein Benutzer von 4D Server. Von daher wird unter angemeldete Anwender "1" angezeigt.

Serverprozeduren individuell starten und beenden

Die Projektmethode START SP SERVICES sieht folgendermaßen aus:

```
` Projektmethode START SP SERVICES
◊vlSPServices:=Execute on server("SP SERVICES";32*1024;"SP SERVICES";*)
```

Da die Funktion Execute on server wie New process arbeitet, wenn sie auf dem Server-Rechner aufgerufen wird, kann auf dem Server-Rechner bzw. jedem Client-Rechner dieselbe Methode (START SP SERVICES) verwendet werden, um die Methode SP SERVICES als Serverprozedur auf dem Server-Rechner aufzurufen.

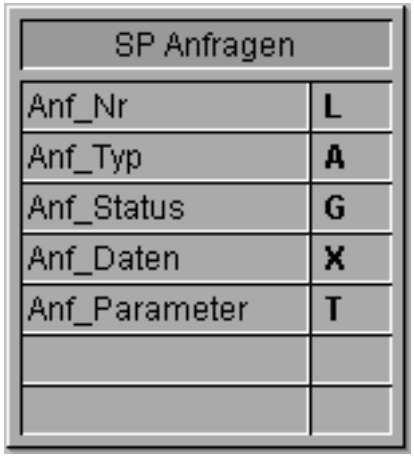
Die Projektmethode STOP SP SERVICES stoppt die Projektmethode SP SERVICES

```
` Projektmethode STOP SP SERVICES
SET PROCESS VARIABLE(◊vlSPServices;vbStopSPServices;True)
```

Startet die Projektmethode SP SERVICES, setzt sie die Prozessvariable vbStopSPServices auf Falsch und durchläuft die Schleife, bis diese Variable vom Typ Boolean den Wert Wahr annimmt. Mit dem Befehl SET PROCESS VARIABLE lässt sich in jedem Benutzerprozess auf dem Server-Rechner bzw. jedem Client-Rechner der Wert der Variablen vbStopSPServices verändern und so die Serverprozedur nach eigenen Wünschen stoppen.

Mit Serverprozeduren kommunizieren

Die Serverprozedur muss Anfragen des Clients jederzeit und in beliebiger Reihenfolge asynchron empfangen und beantworten können. Das lässt sich am einfachsten mit einer Tabelle bewerkstelligen.



Die Tabelle [SP Anfragen] enthält folgende Datenfelder:

• [SP Anfragen]Anf_Nr arbeitet mit der Funktion Sequence number. Es dient nur zum Identifizieren der Anfragen.
• [SP Anfragen]Anf_Typ beschreibt die Art der Anfrage.

• [SP Anfragen]Anf_Status kann folgende Werte annehmen:

Wert	Beschreibung
1	Die Anfrage wurde gesendet, jedoch noch nicht bearbeitet
0	Die Anfrage wurde erfolgreich bearbeitet.
< 0	Die Anfrage wurde bearbeitet, es ist jedoch ein Fehler aufgetreten.

Hinweis: Die Werte wurden willkürlich für dieses Beispiel gewählt, sie sind nicht von 4D vorgegeben.

• [SP Anfragen]Anf_Daten ist ein BLOB mit den Daten der Anfrage. Sie kann sowohl Daten enthalten, die der Anfrager sendet als auch Daten, die die Serverprozedur an den Anfrager zurückgibt.
• [SP Anfragen]Anf_Parameter enthält optional Parameterwerte, die der Anfrager an die Serverprozedur sendet.

Warum eine Tabelle verwenden?

Die Kommunikation zwischen einem Client-Rechner und einer Serverprozedur lässt sich mit den Befehlen GET PROCESS VARIABLE, SET PROCESS VARIABLE und VARIABLE TO VARIABLE herstellen. Diese Befehle werden z. B. im Abschnitt Import mit Serverprozeduren (Beispiel) und in der Projektmethode STOP SP SERVICES verwendet.

Hier muss das System jedoch zulassen, dass die Serverprozedur variable Datenmengen empfängt und zurücksendet. Sie könnten zwar auch Arrays, einschließlich Arrays vom Typ Text und Bild verwenden, zwei Gründe sprechen jedoch für die Verwendung einer Tabelle:

• Der Algorithmus zum Bearbeiten von Anfragen via Datensätzen lässt sich einfacher einrichten. Wollen Sie eine Anfrage von einem Client-Rechner aus senden, müssen Sie in der Tabelle lediglich eine Anfrage hinzufügen. Wird die Anfrage von der Serverprozedur aus beantwortet, muss diese Anfrage lediglich geändert werden.

• Anfragen in einer Tabelle werden auch auf der Festplatte gespeichert. Von daher ist die Größe einer umfangreichen Anfrage kein Problem, da sie aus dem Speicher entfernt werden kann (im Gegensatz zu Daten, die in Arrays gespeichert sind)

Eine Anfrage vom Client-Rechner senden

Die Projektmethode Client post request ist eine generische Methode zum Senden einer Anfrage:

```
    ` Projektmethode Client post request
    ` Client post request ( String { ; Text } ) -> Long
    ` Client post request (Anfrage_Typ { ; Parameter } ) -> Anfrage_Nr
CREATE RECORD([SP Anfragen])
[SP Anfragen]Anf_Nr:=Sequence number([SP Anfragen])
[SP Anfragen]Anf_Typ:=$1
[SP Anfragen]Anf_Status:=1
If (Count parameters>=2)
   [SP Anfragen]Anf_Parameter:=$2
End if
SAVE RECORD([SP Anfragen])
$0:=[SP Anfragen]Anf_Nr
```

Die Methode gibt die Nummer der Anfrage zurück. Die Funktion Sequence number stellt ihre Einzigartigkeit sicher. Wurde der Datensatz in der Datenbank [SP Anfragen] hinzugefügt, kann der Client das Datenfeld [SP Anfragen]Anf_Status befragen und abwarten, bis die Serverprozedur die Anfragen vollständig bearbeitet hat.

Den Status der Anfrage abfragen und das Ergebnis auf dem Client-Rechner erhalten

Die Projektmethode Client get result ist eine generische Methode zum Abfragen des Status der Anfrage. Wie bereits erwähnt, weiss der Client, sobald das Datenfeld [SP Anfragen]Anf_Status einen anderen Wert als 1 hat, dass die Serverprozedur die Anfrage bearbeitet hat (erfolgreich oder nicht erfolgreich).

```
    ` Projektmethode Client get result
    ` Client get result ( Long ; ->BLOB {; Long } ) -> Long
    ` Client get result ( Anfrage-Nr; ->Daten {; Delay } ) -> Error Code
C_LONGINT($0;$1;$vlDelay)
$0:=1
$vlDelay:=0
If (Count parameters>=3)
   $vlDelay:=$3
End if
READ ONLY([SP Anfragen])
Repeat
   QUERY([SP Anfragen];[SP Anfragen]AnfID=$1)
   If (Records in selection([SP Anfragen])>0)
```

```
            If ([SP Anfragen]Anf_Status # 1)
               $2->:=[SP Anfragen]Anf_Daten
               READ WRITE([SP Anfragen])
               While (Locked([SP Anfragen]))
                  WAITING LOOP ($vlDelay)
                  LOAD RECORD([SP Anfragen])
               End while
               DELETE RECORD([SP Anfragen])
               $0:=[SP Anfragen]Anf_Status
            End if
         Else
               ` Datensatz für die Anfrage ging verloren!
               ` Das sollte zwar nicht passieren. Setze trotzdem den Fehler auf -2
                                                                  (willkürlicher Wert)
            $0:=-2
         End if
            ` Die Anfrage wurde noch nicht bearbeitet
         If ($0=1)
            WAITING LOOP ($vlDelay)
         End if
      Until ($0 # 1)
      READ ONLY([SP Anfragen])
```

Hat die Serverprozedur die Anfrage erfolgreich bearbeitet, kopiert die Methode das Ergebnis (falls vorhanden) aus dem Datensatz in das BLOB, dessen Zeiger als Parameter übergeben wurde. Die aufrufende Methode analysiert und verwendet dann die BLOB Daten je nach Art der Anfrage. Beachten Sie, dass der Client für das Löschen des Datensatzes [SP Anfragen] zuständig ist, sobald die Anfrage vollständig bearbeitet ist.

Die kleine Projektmethode WAITING LOOP durchläuft die Schleife, bis eine Anzahl Ticks vergangen ist:

```
      `Projektmethode WAITING LOOP
      ` WAITING LOOP ( Long )
      ` WAITING LOOP ( Delay in ticks )
   C_LONGINT($1)
   $vlStartTicks:=Tickcount
   Repeat
      IDLE
   Until ((Tickcount-$vlStartTicks)>=$1)
```

Zu Erinnerung: Der Befehl DELAY PROCESS hat im Benutzerprozess keine Auswirkung. Mit der Projektmethode WAITING LOOP wartet der Prozess die erforderliche Zeitspanne ab, selbst wenn die Anfrage von einem Benutzerprozess auf einem Client-Rechner ausging.

Serverprozeduren und dazugehörige Unterroutinen

Die Projektmethode SP SERVICES läuft als Serverprozedur auf dem Server-Rechner. Der gesamte Aufbau dieser Methode, hier gezeigt in Pseudocode, ist denkbar einfach:

```
Initialisiere eine Variable “stop”
Wiederhole
   Suche nach Anfragen mit dem Wert des Datenfeldes
                                        [SP Anfragen]Anf_Status ist gleich 1
   Für jede Anfrage
      Rufe je nach Art der Anfrage eine Unterroutine auf
         die das Ergebnis im Datenfeld [SP Anfragen]Anf_Daten speichert
      Ändere den Status der Anfrage, so dass der Client weiß, was passiert ist
   Ende für
   “Schlafe” ein wenig vor erneutem Starten
Bis die Variable “stop” den Wert wahr hat.
```

Hier ist der echte Source Code:

```
   ` Projektmethode SP SERVICES
   ` Die Serverprozedur startet
vbStopSPServices:=False
   ` Die Serverprozedur benötigt auf Tabellen etc. keinen Zugriff im Lese/Schreibmodus
READ ONLY(*)
   ` ...außer der Tabelle [SP Anfragen]
READ WRITE([SP Anfragen])
Repeat
      ` Suche nach noch nicht bearbeiteten Anfragen
   QUERY([SP Anfragen];[SP Anfragen]Anf_Status=1)
      ` Bearbeite diese Anfragen eine nach der anderen
   For ($vlRecord;1;Records in selection([SP Anfragen]))
         ` Ist der Datensatz Anfrage gesperrt, warte bis er freigegeben wird
      While (Locked([SP Anfragen]))
            ` Warte eine Sekunde vor erneutem Versuch
         DELAY PROCESS(Current process;60)
            ` Versuche, Zugriff im Lese/Schreibmodus zu erhalten
         LOAD RECORD([SP Anfragen])
      End while
         ` Nimm an, dass die Anfrage erfolgreich bearbeitet wird
      [SP Anfragen]Anf_Status:=0
```

```
            Case of
               : ([SP Anfragen]Anf_Typ="Server Information")
                  SP DO SERVER INFORMATION
               : ([SP Anfragen]Anf_Typ="Volume List")
                  SP DO VOLUME LIST
               : ([SP Anfragen]Anf_Typ="Browse Directory")
                  SP DO BROWSE DIRECTORY ([SP Anfragen]Anf_Parameter)
                  ` ...
                  ` HIER KÖNNEN ANDERE ANFRAGETYPEN HINZUGEFÜGT WERDEN!
                  ` ...
               Else
                     ` Der Anfragetyp ist unbekannt, gibt Fehler -1 zurück
                                                                  (willkürlicher Wert)
                  [SP Anfragen]Anf_Status:=-1
            End case
               ` Sorge dafür,dass der Anfragestatus ungleich 1 ist
               ` (falls eine Unterroutine ihn auf 1 setzt)
            If ([SP Anfragen]Anf_Status=1)
               [SP Anfragen]Anf_Status:=-3
            End if
               ` Aktualisiere den Datensatz Anfrage
            SAVE RECORD([SP Anfragen])
               ` Gehe zur nächsten unbearbeiteten Anfrage
            NEXT RECORD([SP Anfragen])
         End for
            ` Gib den zuletzt bearbeiteten Datensatz Anfrage frei
         UNLOAD RECORD([SP Anfragen])
            ` Warte eine Sekunde vor erneutem Starten der Antwort auf die Anfrage
         DELAY PROCESS(Current process;60)
            ` Bleibe in der Schleife, bis die SP angewiesen wird, die Ausführung zu stoppen
      Until (vbStopSPServices)
```

Die Projektmethode SP SERVICES kann als Vorlage zur Integration neuer Services in einer Datenbank dienen. In diesem Abschnitt betrachten wir näher die Unterroutinen SP DO SERVER INFORMATION und SP DO VOLUME LIST. Die Unterroutine SP DO BROWSE DIRECTORY, die den Parameter übernimmt, den der Client im Datenfeld [SP Anfragen]Anf_Parameter übersendet, wird hier nicht behandelt.

Die Projektmethode SP SERVICES ruft je nach Anfragetyp eine Unterrroutine auf, welche die resultierenden Daten im Datenfeld [SP Anfragen]Anf_Daten speichert. SP SERVICES sichert den Datensatz und ändert den Status der Anfrage.

Hier folgt die Unterroutine SP DO SERVER INFORMATION. Sie speichert die server-bezogene Information im BLOB. Eine andere Projektmethode entnimmt die BLOB Daten entsprechend auf dem Client-Rechner.

```
` Projektmethode SP DO SERVER INFORMATION
TEXT TO BLOB(Application version(*);[SP Anfragen]Anf_Daten;Pascal string)
TEXT TO BLOB(Structure file;[SP Anfragen]Anf_Daten;Pascal string;*)
TEXT TO BLOB(Data file;[SP Anfragen]Anf_Daten;Pascal string;*)
PLATFORM PROPERTIES($vlPlatform;$vlSystem;$vlMachine)
VARIABLE TO BLOB($vlPlatform;[SP Anfragen]Anf_Daten;*)
VARIABLE TO BLOB($vlSystem;[SP Anfragen]Anf_Daten;*)
VARIABLE TO BLOB($vlMachine;[SP Anfragen]Anf_Daten;*)
```

Hier folgt die Unterroutine SP DO VOLUME LIST. Sie speichert die volume-bezogene Information im BLOB. Eine andere Projektmethode entnimmt die BLOB Daten entsprechend auf dem Client Rechner.

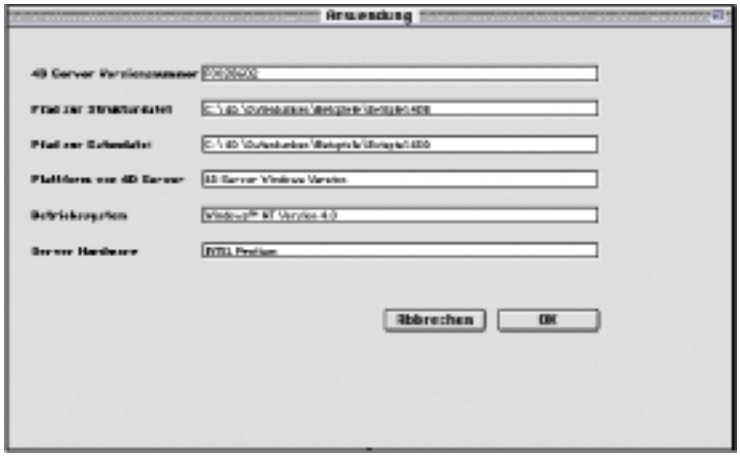
```
` Projektmethode SP DO VOLUME LIST
VOLUME LIST($asVName)
$vlSize:=Size of array($asVName)
ARRAY REAL($arVSize;$vlSize)
ARRAY REAL($arVUsedSpace;$vlSize)
ARRAY REAL($arVFreeSpace;$vlSize)
For ($vlElem;1;$vlSize)
   VOLUME ATTRIBUTES($asVName{$vlElem};$arVSize{$vlElem};
      $arVUsedSpace{$vlElem};$arVFreeSpace{$vlELem})
End for
VARIABLE TO BLOB($asVName;[SP Anfragen]Anf_Daten)
VARIABLE TO BLOB($arVSize;[SP Anfragen]Anf_Daten;*)
VARIABLE TO BLOB($arVUsedSpace;[SP Anfragen]Anf_Daten;*)
VARIABLE TO BLOB($arVFreeSpace;[SP Anfragen]Anf_Daten;*)
```

Die Server Information auf einem Client-Rechner zeigen

Mit den generischen Projektmethoden Client post request und Client get result zeigt die Projektmethode M_SERVER_INFORMATION die Serverinformation an, die die Serverprozedur auf dem Client-Rechner zurückgibt. Diese Methode kann einem Menübefehl zugeordnet werden oder z.B. über eine Objektmethode für eine Schaltfläche aufgerufen werden:

```
   ` M_SERVER_INFORMATION
C_BLOB(vxData)
C_LONGINT($vlAnf_Nr;$vlErrCode;$vlOffset)
   ` Stelle die Anfrage
$vlAnfID:=Client post request ("Server Information")
   ` Frage den Anfragestatus ab und erhalte das Ergebnis
$vlErrCode:=Client get result ($vlAnfID;->vxDaten;60)
   ` Ist die Anfrage erfolgreich abgeschlossen, zeige das Ergebnis an
If ($vlErrCode=0)
      ` Entnehme die Information aus dem BLOB
   $vlOffset:=0
   vsServerVersion:=BLOB to text(vxDaten;Pascal string;$vlOffset)
   vsStructureFile:=BLOB to text(vxDaten;Pascal string;$vlOffset)
   vsDataFile:=BLOB to text(vxDaten;Pascal string;$vlOffset)
   BLOB TO VARIABLE(vxDaten;$vlPlatform;$vlOffset)
   BLOB TO VARIABLE(vxDaten;$vlSystem;$vlOffset)
   BLOB TO VARIABLE(vxDaten;$vlMachine;$vlOffset)
      ` Analysiere die Plattform Eigenschaften
   vs4DPlatform:="Unbekannte 4D Server Version"
   vsSystem:="Unbekannte System Version"
   vsMachine:="Unbekannter Rechner"
      `...
      ` Hier ist der Code (nicht aufgelistet) der $vlSystem und $vlMachine analysiert
      ` (siehe Beispiel für den Befehl PLATFORM PROPERTIES)
      ` ...
      ` Zeige das Ergebnis
   DIALOG([SP Anfragen];"SERVER INFORMATION")
Else
   ALERT("Anfrage error "+String($vlErrCode))
End if
   ` BLOB wird nicht länger benötigt
CLEAR VARIABLE(vxDaten)
```

Hier ist das Formular [SP Anfragen];"SERVER INFORMATION" in der Benutzer- oder Runtime-Umgebung:



In diesem Fenster wird die Information von einem 4D Server-Rechner für Windows NT auf einem Client-Rechner für Macintosh angezeigt.

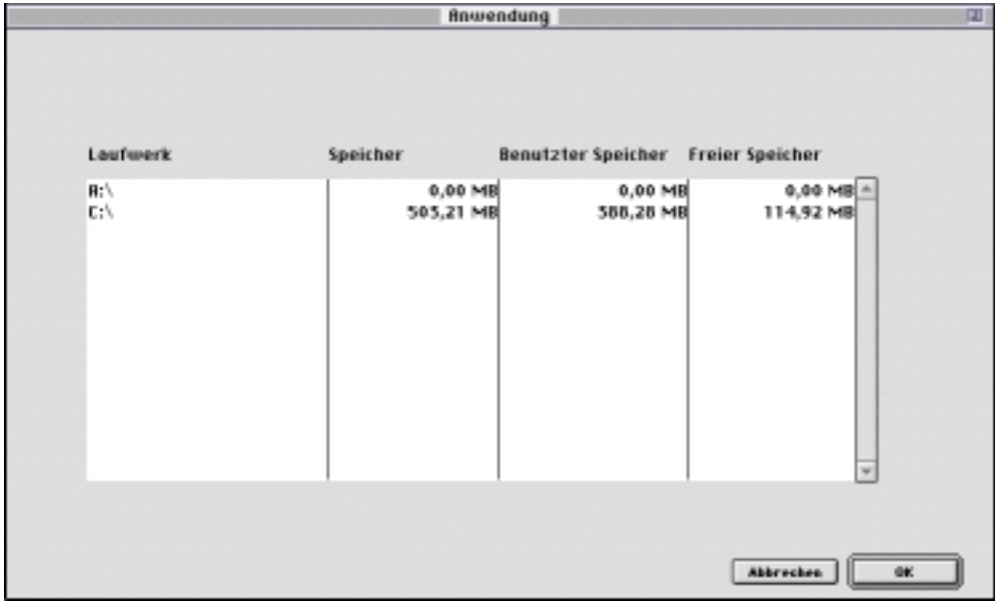
Laufwerke des Server-Rechners auf einem Client-Rechner anzeigen

Mit den generischen Projektmethoden Client post request und Client get result zeigt die Projektmethode M_SERVER_VOLUMES auf dem Client-Rechner die Laufwerke des Server-Rechners an, die die Serverprozedur zurückgibt. Diese Methode kann einem Menübefehl zugeordnet werden oder z.B. über eine Objektmethode für eine Schaltfläche aufgerufen werden:

```
   ` M_SERVER_VOLUMES
C_BLOB(vxDaten)
   ` Stelle die Anfrage
$vlReqID:=Client post request ("LAUFWERK")
   ` Frage den Anfragestatus ab und erhalte das Ergebnis
$vlErrCode:=Client get result ($vlAnf_Nr;->vxDaten;120)
   ` Ist die Anfrage erfolgreich abgeschlossen, zeige das Ergebnis an
If ($vlErrCode=0)
      ` Entnehme die Information aus dem BLOB
   $vlOffset:=0
   BLOB TO VARIABLE(vxDaten;asVName;$vlOffset)
   BLOB TO VARIABLE(vxDaten;arVSize;$vlOffset)
   BLOB TO VARIABLE(vxDaten;arVUsedSpace;$vlOffset)
```

```
    BLOB TO VARIABLE(vxDaten;arVFreeSpace;$vlOffset)
    For ($vlElem;1;Size of array(arVSize))
          ` Konvertiere von bytes in MB
       arVSize{$vlElem}:=arVSize{$vlElem}/1048576
       arVUsedSpace{$vlElem}:=arVUsedSpace{$vlElem}/1048576
       arVFreeSpace{$vlElem}:=arVFreeSpace{$vlElem}/1048576
    End for
          ` Zeige das Ergebnis
    DIALOG([SP Anfragen];"LAUFWERK")
Else
    ALERT("Anfrage error "+String($vlErrCode))
End if
    ` BLOB wird nicht länger benötigt
CLEAR VARIABLE(vxDaten)
```

Hier ist das Formular [SP Anfragen];"LAUFWERK" in der Benutzer- oder Runtime-Umgebung:



In diesem Fenster wird die Information von einem 4D Server-Rechner für Windows auf einem Client-Rechner für MacOS angezeigt.

Referenz

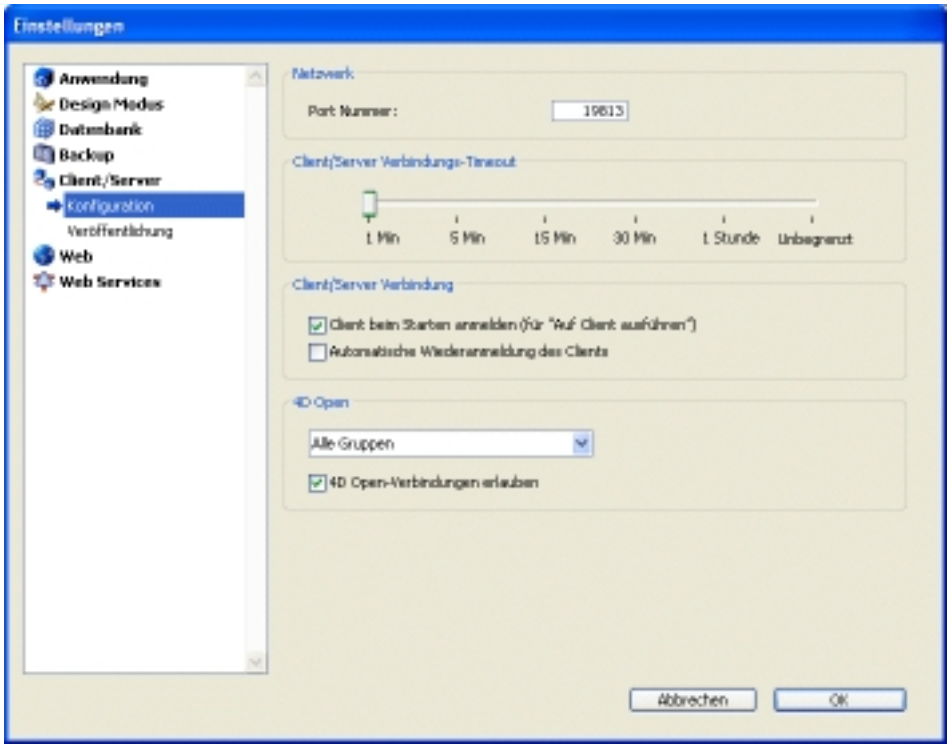
BLOB Befehle, Execute on server, Import mit Serverprozeduren (Beispiel), Serverprozeduren.

Sie können Serverprozeduren auf einem oder mehreren anderen Clients ausführen. Sie arbeiten wie die Serverprozeduren auf dem Server. Der einzige Unterschied ist, dass eine auf 4D Client ausgeführte Serverprozedur auch die Dateneingabe zulässt. Weitere Informationen dazu finden Sie im Abschnitt Serverprozeduren.

Darüberhinaus muss jede Arbeitsstation, die auf Anfrage des Servers oder eines anderen Clients Prozeduren ausführen soll, explizit für die Arbeitssitzung registriert sein. Dafür gibt es zwei Möglichkeiten: Jeden Client automatisch beim Anmelden registrieren oder Clients per Programmierung registrieren.

Jeden 4D Client automatisch beim Anmelden an 4D Server registrieren

Wählen Sie dazu in den Einstellungen der Datenbank unter dem Thema Client-Server auf der Seite **Konfiguration** die Option "Client beim Starten registrieren":



Ist diese Option markiert, gilt jeder 4D Client, der sich an die Datenbank anmeldet, in 4D Server als Client, der Serverprozeduren ausführen kann. Auf dem Server wird in der Gruppe Benutzerprozesse ein Prozess mit dem Namen des Client Rechners angelegt.

4D Client per Programmierung registrieren

Sie können auch einen oder mehrere 4D Clients per Programmierung registrieren. In diesem Fall wählen Sie die Arbeitsstationen, die Sie registrieren wollen und ihre Namen selbst. Verwenden Sie dazu den Befehl REGISTER CLIENT.

Registrierung von 4D Client aufheben

Unabhängig von der Art der Registrierung können Sie jede Registrierung der Arbeitsstation für die aktuelle Sitzung wieder aufheben. Rufen Sie dazu für die betreffende(n) Arbeitsstation(en) den Befehl UNREGISTER CLIENT auf.

Der Prozess, der die Client-Registrierung verwaltet – er hat den Namen des Client – wird dann auf dem Server aus der Gruppe Benutzerprozesse und auf der Arbeitsstation entfernt.

Hinweis: Mit dem Befehl GET REGISTERED CLIENTS erhalten Sie die Liste der für die Sitzung registrierten Clients und ihr jeweiliges Arbeitsaufkommen, d.h. die Anzahl der noch auszuführenden Methoden.

Weitere Informationen zu diesen Befehlen finden Sie im Handbuch *4th Dimension Programmiersprache* im Kapitel Prozesse.